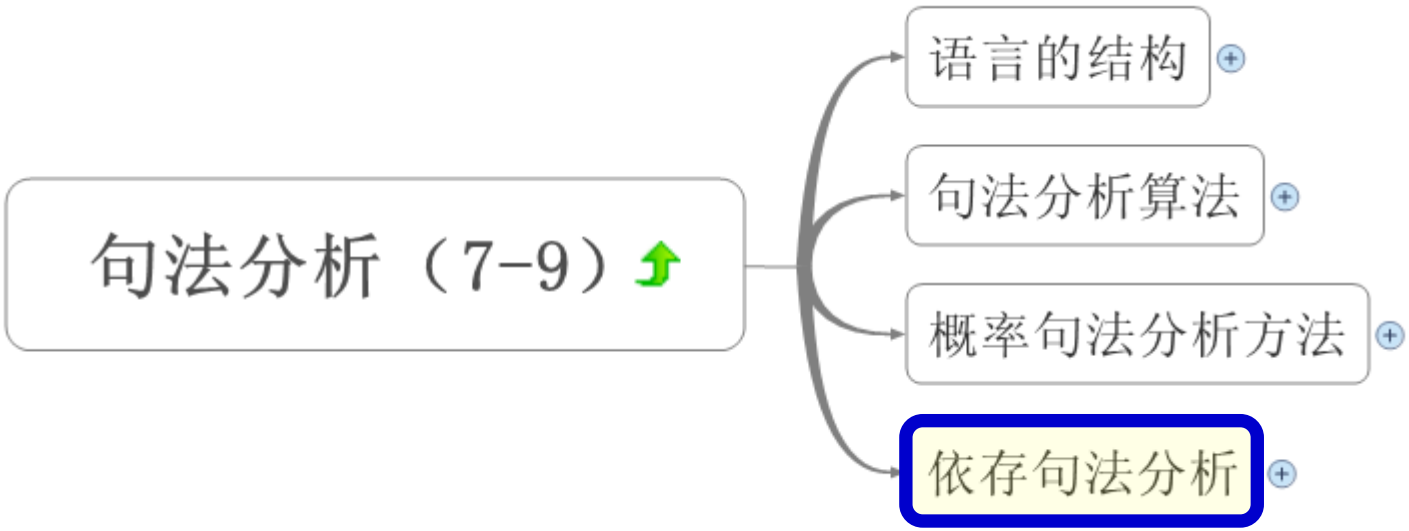
计算语言学

第9讲 句法分析（三）

刘群
中国科学院计算技术研究所
liuqun@ict.ac.cn

中国科学院研究生院2010年春季课程讲义

内容提要

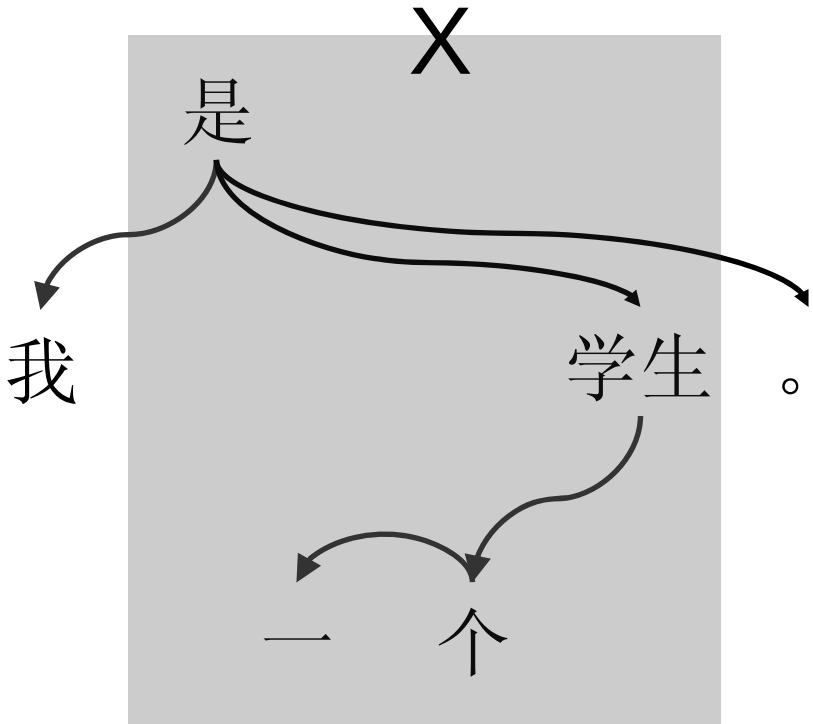


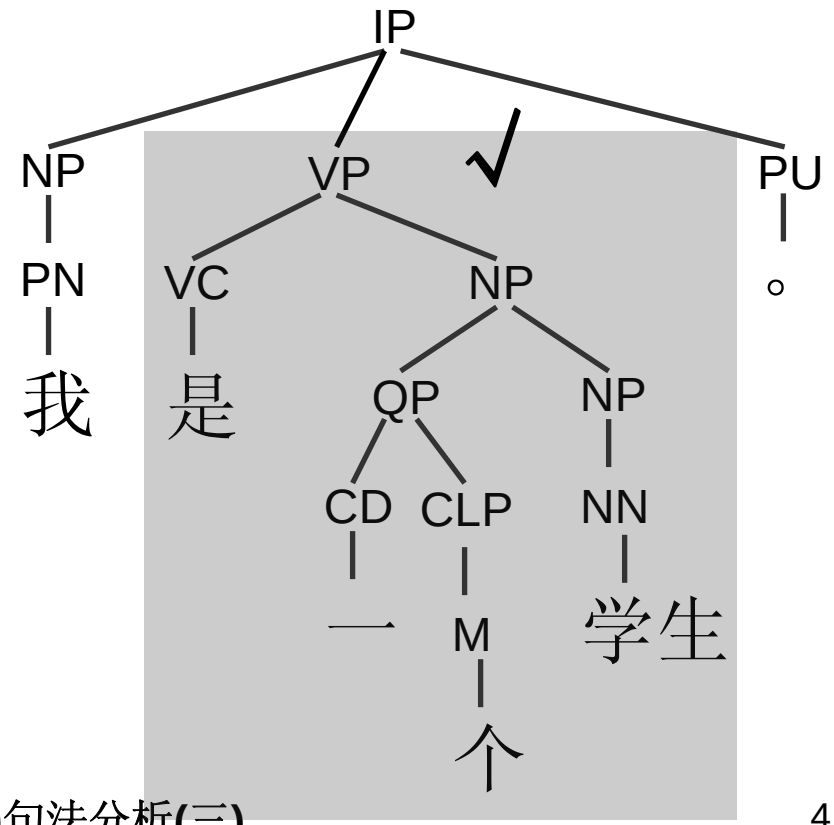
依存分析

- 依存结构和依存语法
- 短语结构树转依存树
- 专门的依存分析模型
  - 概率依存模型
  - 最大生成树模型
  - 状态转移模型

依存分析简介

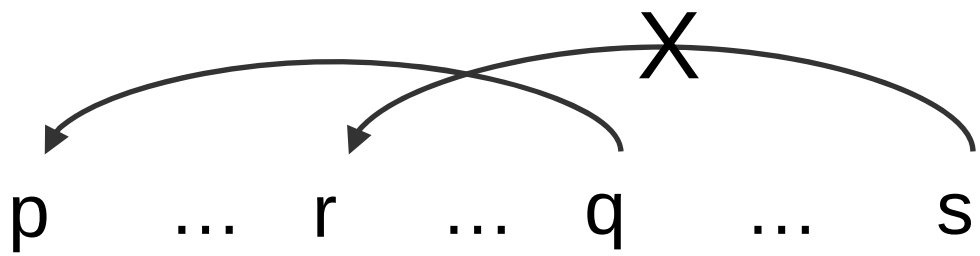
- 依存分析与短语结构分析类似，但有所不同：
  依存分析丢掉了跨度信息和跨度上的句法标识





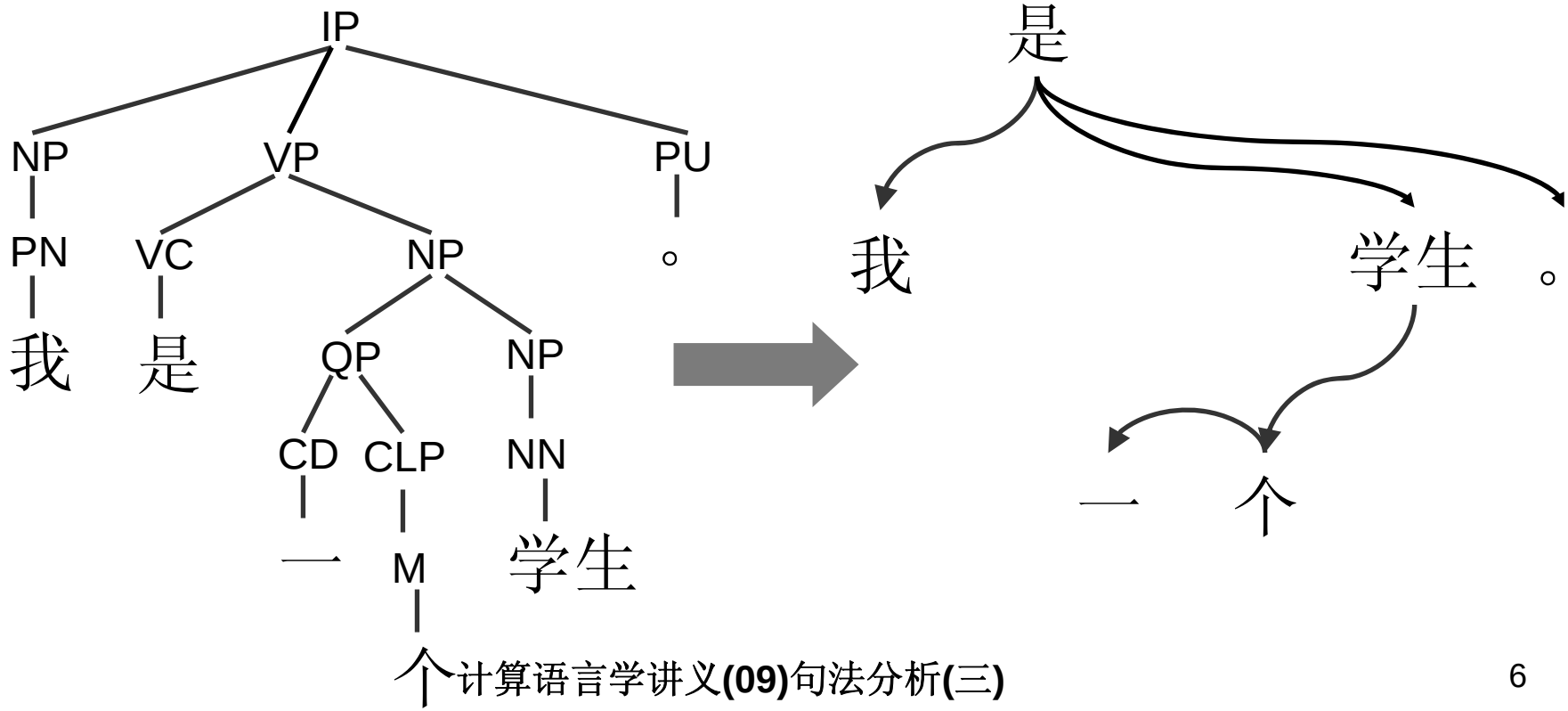
依存分析简介

- 大多数语言，包括汉语和英语，满足投射性。所谓投射性是指：如果词p依存于词q，那么p和q之间的任意词r就不能依存到p和q所构成的跨度之外



短语结构树转依存树

- 任何短语结构树句法分析模型输出的句法树，通过 Yamada and Matsumoto (2003) 的中心词映射规则即可转化为依存结构树



短语结构树转依存树

- 中心词映射规则示例
  - 规则： IP right { IP VP }
  - 意义：对于句法树中标识为IP的节点，自右向左扫描该节点的所有孩子，第一个出现在列表{ IP VP }中的孩子即为中心孩子节点。其他孩子节点的中心词将依存到中心孩子节点的中心词
- 对于给定的短语结构树，自底向上应用中心词映射规则，即可确定各词之间的依存关系

依存分析模型

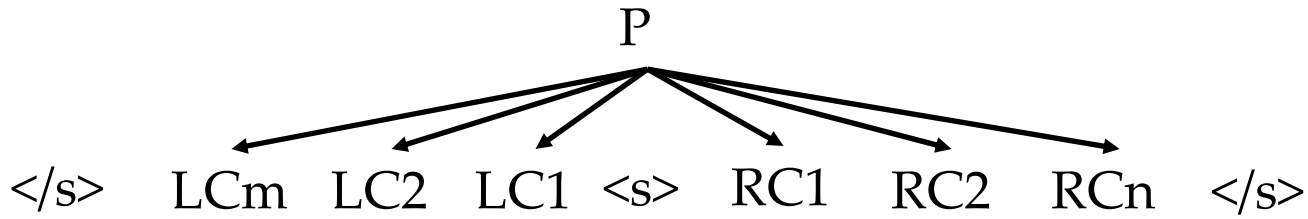
- 生成式依存模型
  - 词汇依存概率模型（Collins模型）
  - 依存生成概率模型（Eisner模型）
- 判别式依存模型
  - 状态转移模型
  - 最大生成树模型

词汇依存概率模型

- Collins, 1996
- 训练：
  通过极大似然估计，在树库中统计出任意两个词之间存在特定依存关系的概率
- 解码：
  寻找使得所有依存词对的依存概率的乘积最大的依存树

概率依存模型
(Eisner, 1996)

- 给定输入语句的一棵可能的依存树，设该树中任一节点P，它的左孩子由近及远分别为LC1，LC2，…，LCm；右孩子分别为RC1， RC2， …， RCn



- 定义P生成其所有孩子的概率为：

$$\mathrm{Gen}(P)=\prod_{i=1}^{m}\Pr(LC_i.word \mid LC_{i-1}.POS, P.word)$$

$$\times\prod_{i=1}^{n}\Pr(RC_i.word \mid RC_{i-1}.POS, P.word)$$

概率依存模型
(Eisner, 1996)

- 对于每棵候选依存树T，整棵树的生成概率定义为树中所有节点生成概率的乘积

$$Gen(T) = \prod_{x \in T} Gen(x)$$

- 解码的任务就是寻找生成概率最大的依存树

最大生成树模型

- McDonald et al., 2005
- McDonald and Pereira, 2006
- 给定一个包含N个词的句子，任意两个词之间都可能存在依存关系，共有N*(N-1)种可能的依存边(不能含有依存到自己的自环)，只是依存强弱不同
- 将依存强弱表示为这个完全图中边的分数。于是，寻找最可能的依存树的任务就转化为寻找这个完全图的最大生成树

最大生成树模型

- 每条边p→c的分数定义为

$$\text{score}(p \rightarrow c) = f(p \rightarrow c) \cdot w$$

- f()函数返回依存边p→c的特征向量；w为权重向量，它由判别式训练得到

最大生成树模型—解码搜索

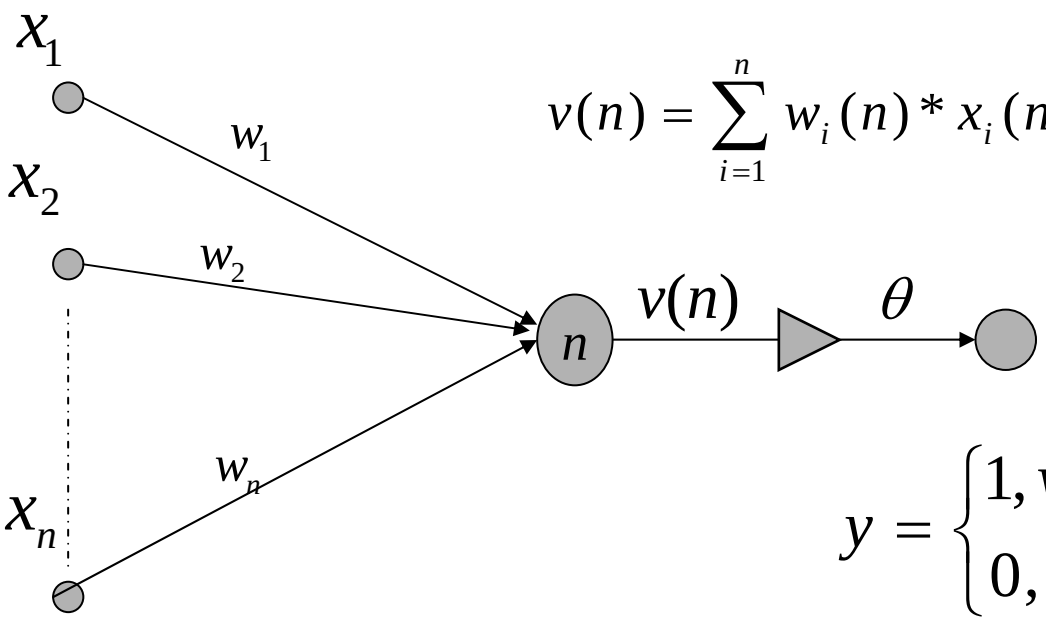
- Chu-Liu-Edmonds算法：最大生成树算法

感知机简介

- 感知机是是一种双层神经网络模型，一层为输入层，另一层具有计算单元，可以通过监督学习建立模式判别的能力，在判别训练中广泛应用。
- 学习的目标是通过改变权值使神经网络由给定的输入得到给定的输出。
- 用于解决二值分类问题。

感知机简介

感知机模型示意图：



$$v(n) = \sum_{i=1}^{n} w_i(n) * x_i(n) = W^T(n) * X(n)$$

$$y = \begin{cases} 1, v(n) > \theta \\ 0, otherwise \end{cases}$$

在线性可分的情况下，上面的$v(n)$就是分类的超平面方程。

自然语言处理中的感知机

- 自然语言处理处理中，我们通常用感知机解决序列标注问题
- 在序列标注问题中，标注通常都不只有两个，不能直接用原始的感知机解决

自然语言处理中的感知机

- 把条件和标注放在一起，定义一个特征向量
- 对每一个标注，与条件组合，得到一个特征项目
- 标注的选择方法之一：
  - 对于每一个标注，用一个感知机判断结果是0还是1，0解释为错误标注，1解释为正确标注。
  - 问题：如果有多个标注判断为正确，怎么办？
- 标注的选择方法之一：
  - 采用无阈值的感知机，对每一个特征向量，只计算出$v(n)$，取$v(n)$最大的标注作为最优标注

感知机应用

- 传统的感知机算法主要应用在两类的分类问题上
- 当前，在自然语言处理方面有如下应用：
  - 基于HMM的词性标注(Michael Collins)
  - Incremental Parsing (Michael Collins)
  - 语言模型的训练(于浩，步丰林，高剑峰)
  - Machine Translation (Percy Liang etc.)

感知机的训练规则

- Perceptron 规则
  - 用于有阈值的两类分类情况下
- Delta 规则
  - 用于无阈值的情况下

感知机的训练规则

- Perceptron规则
  - 以分类错误最小化作为优化目标，通过贪心算法求解损失函数的极值

$$MCELoss(\lambda) = \sum_{i=1}^{M} (t_i - O(C_i))$$

感知机的训练规则

- Perceptron 规则的算法

(1) For iter=1 to T

(2) Foreach training_data

N= t_i – O(C_i)

(3) Foreach $c_m \in C_i$

$$\Delta\lambda_m = \eta N c_m, \lambda_m = \lambda_m + \Delta\lambda_m$$

C_i是特征的集合， c_m是特征， t_i 是样本对应的标准输出， O(C_i)是感知机计算出的输出值。

感知机的训练规则

- Delta规则
  - 以分类错误平方最小化作为优化目标，使用梯度下降方法求解损失函数极值

$$MSELoss(\lambda) = \frac{1}{2}\sum_{i=1}^{M}(Score(W_i^R, \lambda) - Score(W_i^*, \lambda))^2$$

  - MSELoss可以用梯度下降的方式求极值，对lamda求偏导

$$\Delta\lambda_m = \frac{\delta MSELoss(\lambda)}{\delta\lambda_m}$$

$$= \sum_{i=1}^{M}(Score(W_i^R, \lambda) - Score(W_i^*, \lambda)) * (f_m(W_i^R) - f_m(W_i^*))$$

感知机的训练规则

- Delta规则

感知机的训练规则

- Delta 规则的算法

(1) $\Delta\lambda_m = 0, m = 1,...D$

(2) for iter=1 to T

(3) foreach training_data

N=$Score(F_i^R, \lambda) - Score(F_i^*, \lambda)$

(4) foreach $c_m \in C_i$

$\Delta\lambda_m = \Delta\lambda_m - \eta N c_m$

(5) $\lambda_m = \lambda_m + \Delta\lambda_m$

感知机的训练规则

- Perceptron规则和Delta规则的主要区别
  - 特征权值更新的策略
    - Perceptron: N= t_i – O(C_i)
    - Delta:

$$\mathrm{N}=Score(F_i^R,\lambda)-Score(F_i^*,\lambda)$$

  - 特征权值更新的时机
    - Perceptron: 每输入一个训练数据就更新一次权值
    - Delta:当输入所有的训练样本之后，在一次迭代结束时更新

感知机算法的两个扩展

- Voted perceptron
  - 训练时与原始感知机训练算法相同，但保留了每次迭代时生成的权重向量，在测试时，每次迭代生成的权重向量都会选出一个分值最高的候选，以得票最多的候选作为转换结果。
- Averaged perceptron
  - 如果用 $\lambda_s^{t,i}$ 表示特征f_s在经过t次迭代之后，在输入第i个训练样本之后的值，则采用参数平均化的方式计算f_s的权值

$$\lambda_s = \sum_{t=1,..T;i=1,..N} \lambda_s^{t,i} / NT$$

感知机算法的两个扩展

- 投票式感知机和参数平均化都是为了避免由于学习速率过大所引起的训练过程中出现的震荡现象。
- 这两种方式在模型参数训练的过程中与原始感知机的算法一样。投票式感知机是在测试时选取“得票”最高的候选，参数平均化则是在测试时利用平均化后的特征权值直接选取分数最高的候选。

最大生成树模型—感知机训练

- 训练集T={(xt, yt), t=1..|T|}
- For n = 1 .. N
  - For t = 1 .. |T|
    - y' = Decode(**w**, xt);
    - **w** = **w** + **F**(xt,yt) – **F**(xt,y')
  - End for
- End for

任何你选用的解码方法

最大生成树模型—平均参数感知机训练

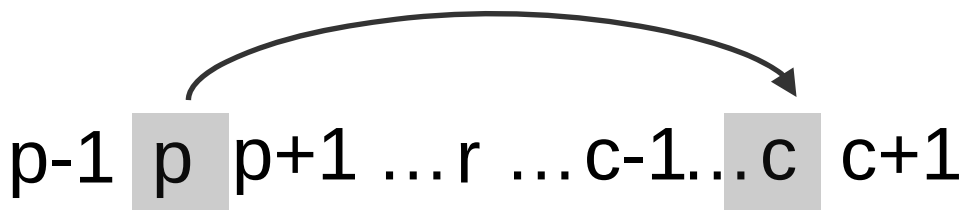
- 训练集T={(xt, yt) , t=1..|T|}
- For n = 1 .. N
  - **v** = 0
  - For t = 1 .. |T|
    - y′ = Decode(**w**, xt);
    - **w** = **w** + **F**(xt,yt) – **F**(xt,y′)
    - **v** = **v** + **w**
  - End for
  - **w_avg** = **v** / (N*|T|)
- End for

最大生成树模型—特征设计

- 特征设计针对边进行，而非节点
- 任意一条p→c的特征可以取那些呢？

最大生成树模型—特征设计

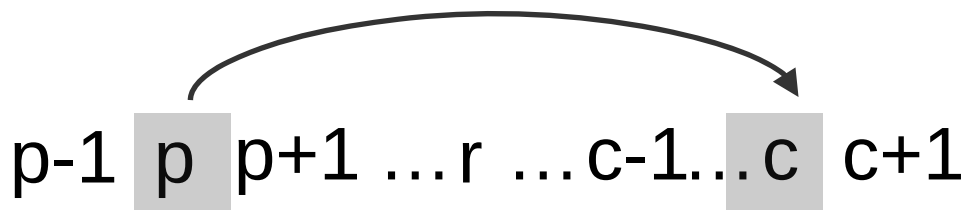
- 一元特征



Pword, Ppos
Pword
Ppos
Cword, Cpos
Cword
Cpos

最大生成树模型—特征设计

- 二元特征



Pword, Ppos, Cword, Cpos

Ppos, Cword, Cpos

Pword, Cword, Cpos

Pword, Ppos, Cpos

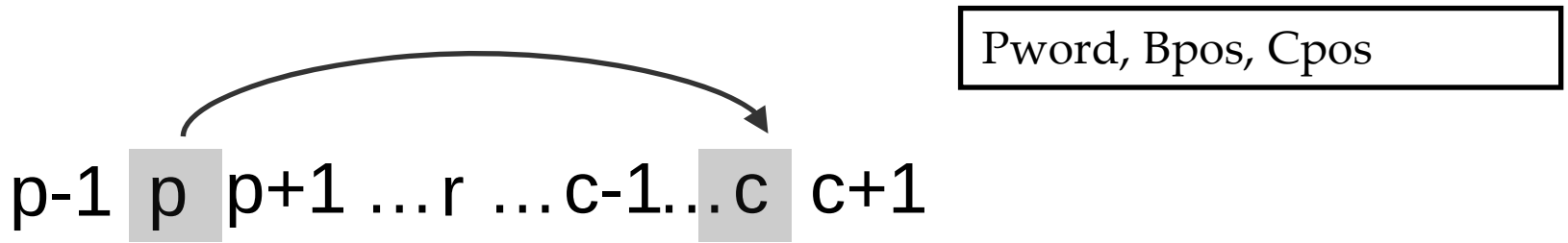
Pword, Ppos, Cword

Pword, Cword

Ppos, Cpos

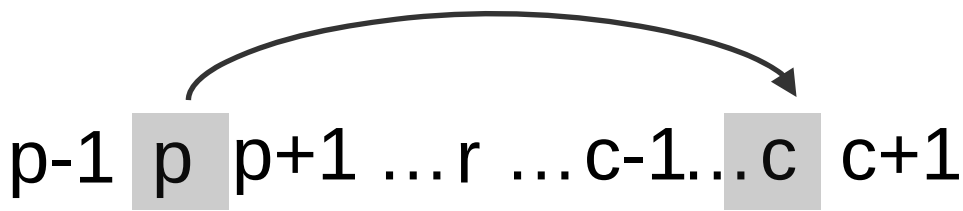
最大生成树模型—特征设计

- 词间词性标注特征，Bpos为父亲p和儿子c之间的一个词的词性标注



最大生成树模型—特征设计

- 父亲儿子周围词性标注特征



Ppos,Ppos+1,Cpos-1, Cpos
Ppos-1, Ppos, Cpos-1, Cpos
Ppos, Ppos+1, Cpos, Cpos+1

最大生成树模型—特征设计

- 除以上特征本身，所有特征都加上父亲-儿子的顺序ord和距离dis，构成一组新的，更细化的特征
- Ord = (Index(p) > Index(c)) ? “Left” : “Right”
- Dis = abs(Index(p) – Index(c))

状态转移模型

- 分析过程的任一时刻称为一个状态，依据该状态下的特征做出某种决策，从而转入新的状态。状态转移的方式有多轮扫描方式和移进规约方式

多轮扫描状态转移模型

- Yamada and Matsumoto, 2003
- 多轮扫描方式：对输入语句的词语序列进行多趟扫描，每趟都试图将存在依存关系的相邻词对(或者子树对)合并为更大的子树，直到形成一棵完整的依存树或者再也没有新的依存对产生

多轮扫描状态转移模型

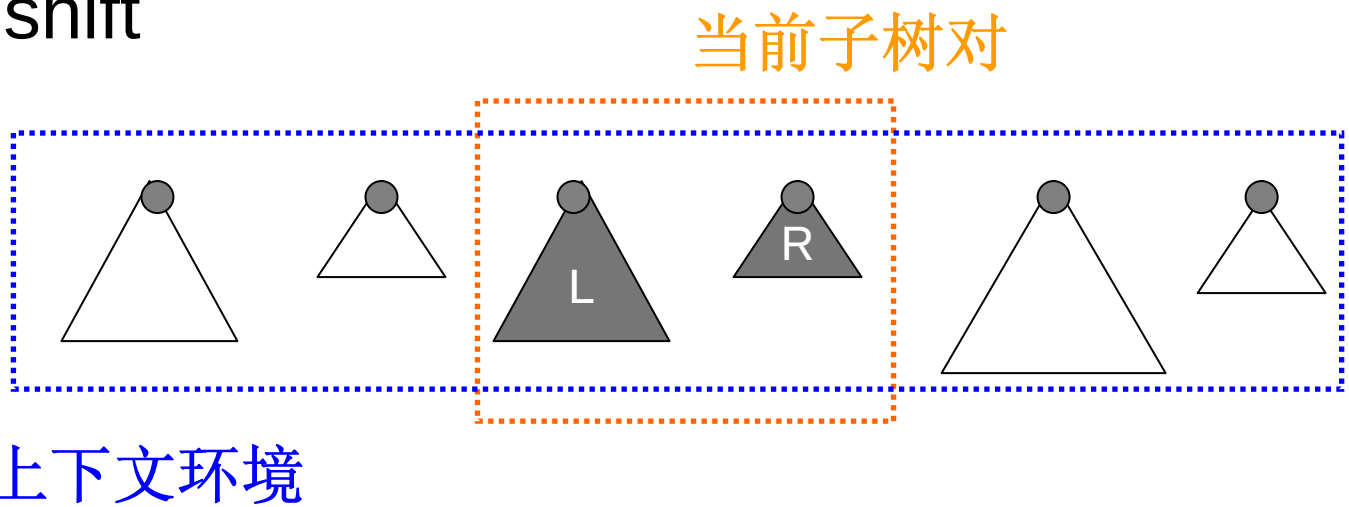
- Yamada and Matsumoto, 2003
- 基本思想
- 状态和转移
- 训练和解码

多轮扫描状态转移模型—基本思想

- 自左到右一趟趟扫描输入串，合并可以合并的相邻子树或词（左依存于右，或反之），直到整个输入串合并为一个依存树或者再也没有相邻子树能够合并为止

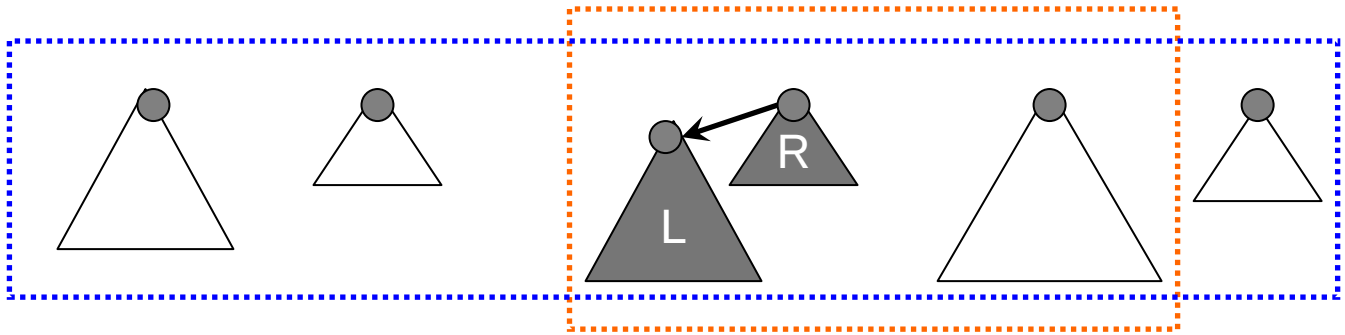
状态和转移

- 在逐趟扫描的过程中，任意时刻的状态是指：当前子树对及它们所处的特定上下文环境
- 任意状态下，三种可能的转移动作：left，right和shift



状态和转移

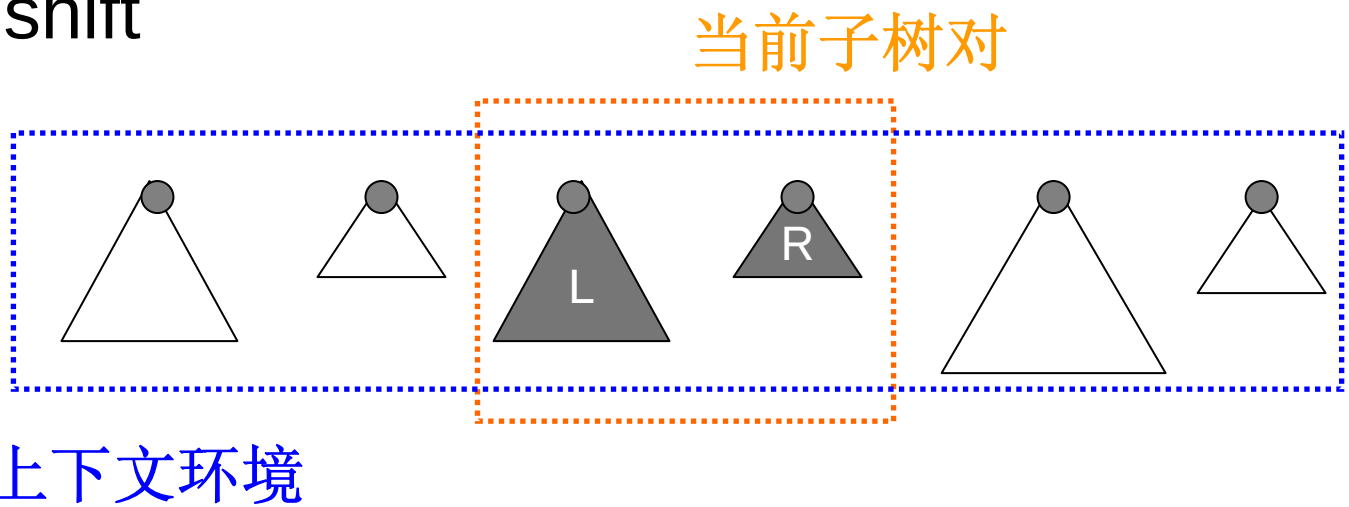
- 在逐趟扫描的过程中，任意时刻的状态是指：当前子树对及它们所处的特定上下文环境
- 任意状态下，三种可能的转移动作：left，right和shift



Left：L←R

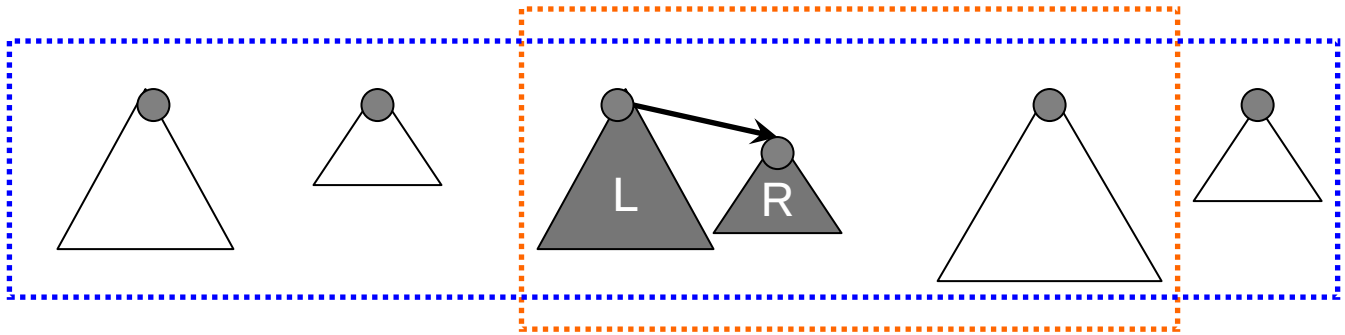
状态和转移

- 在逐趟扫描的过程中，任意时刻的状态是指：当前子树对及它们所处的特定上下文环境
- 任意状态下，三种可能的转移动作：left，right和shift



状态和转移

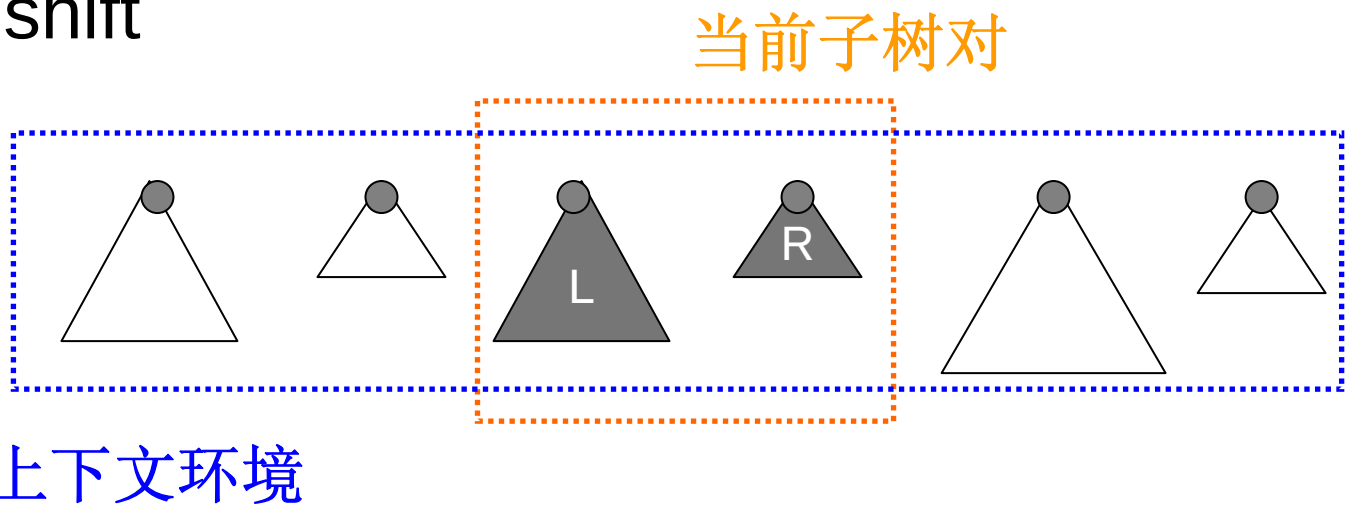
- 在逐趟扫描的过程中，任意时刻的状态是指：当前子树对及它们所处的特定上下文环境
- 任意状态下，三种可能的转移动作：left，right和shift



Right：L→R

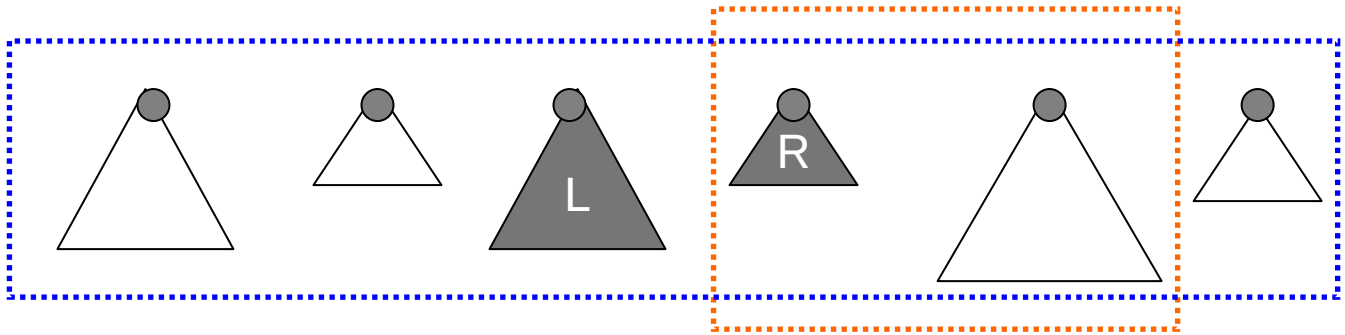
状态和转移

- 在逐趟扫描的过程中，任意时刻的状态是指：当前子树对及它们所处的特定上下文环境
- 任意状态下，三种可能的转移动作：left，right和shift



状态和转移

- 在逐趟扫描的过程中，任意时刻的状态是指：当前子树对及它们所处的特定上下文环境
- 任意状态下，三种可能的转移动作：left，right和shift



Shift

训练和解码

- 训练：
  在树库的训练集中抽取状态转移实例，训练判别式分类器，比如SVM
- 解码：
  在逐趟扫描的过程中，分类器根据当前子树对及其上下文的特征，预测需要执行的状态转移操作并转移到新的状态。这一过程持续至结束条件满足

移进规约状态转移模型

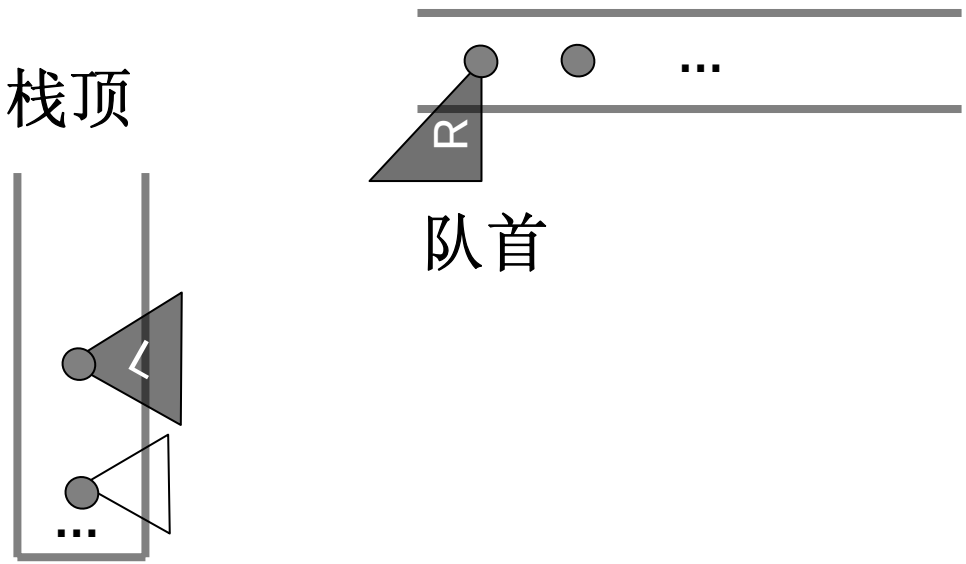
- Nivre et al. , 2006
- 基本思想
- 状态和转移
- 训练和解码
- nbest全局训练

移进归约状态转移模型—基本思想

- 在分析过程中维护一个堆栈和一个队列，堆栈用以存储到目前为止所有的依存子树，队列存储尚未被分析到的词。堆栈顶端和队列的头部确定了当前分析器的状态，依据该状态决定进行移进、规约或者建立栈顶元素与队首元素的依存关系的操作，从而转入新的状态

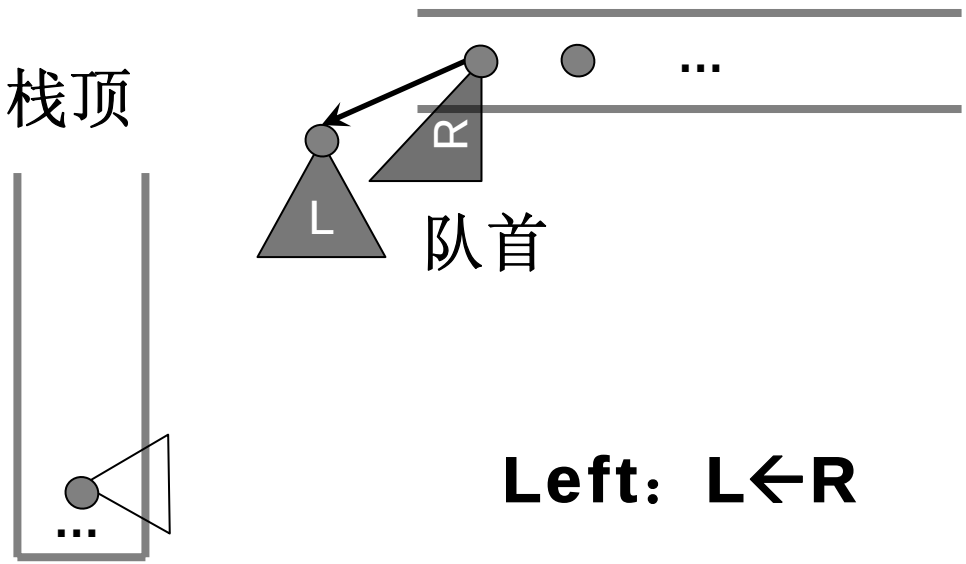
状态和转移

- 在逐趟扫描的过程中，任意时刻的状态是指：当前堆栈和当前队列的格局
- 任意状态下，四种转移动作：left，right，shift和reduce



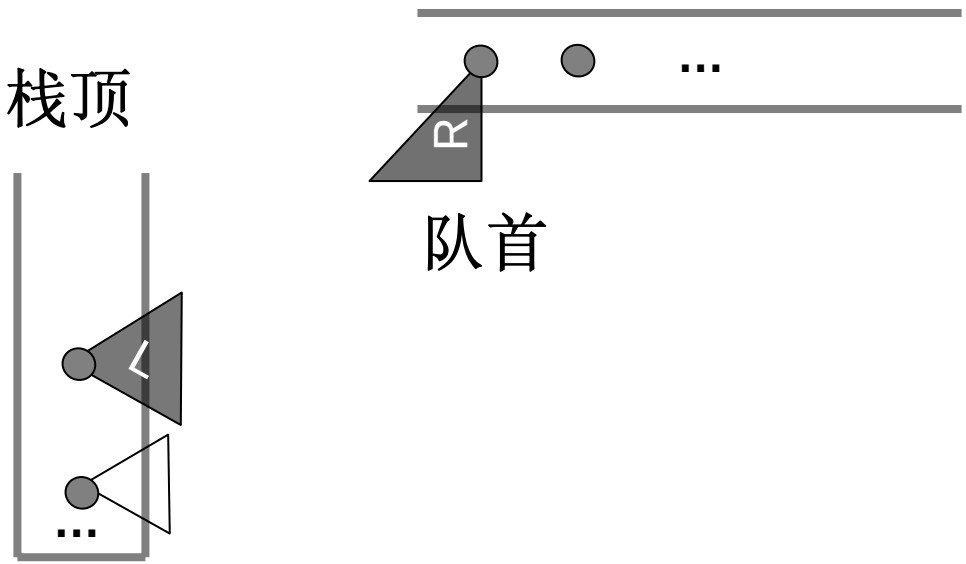
状态和转移

- 在逐趟扫描的过程中，任意时刻的状态是指：当前堆栈和当前队列的格局
- 任意状态下，四种转移动作：left，right，shift和reduce



状态和转移

- 在逐趟扫描的过程中，任意时刻的状态是指：当前堆栈和当前队列的格局
- 任意状态下，四种转移动作：left，right，shift和reduce

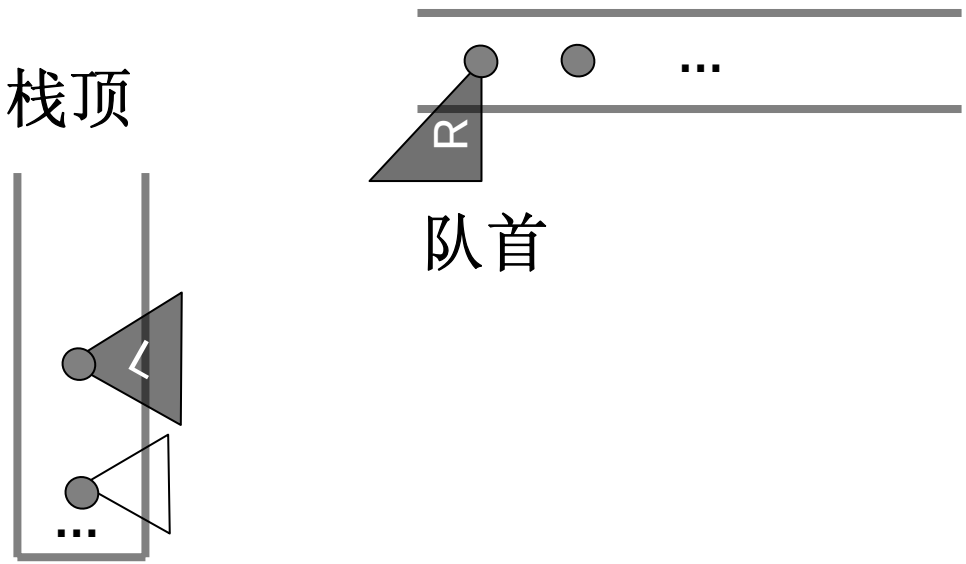


状态和转移

- 在逐趟扫描的过程中，任意时刻的状态是指：当前堆栈和当前队列的格局
- 任意状态下，四种转移动作：left，right，shift和reduce

状态和转移

- 在逐趟扫描的过程中，任意时刻的状态是指：当前堆栈和当前队列的格局
- 任意状态下，四种转移动作：left，right，shift和reduce

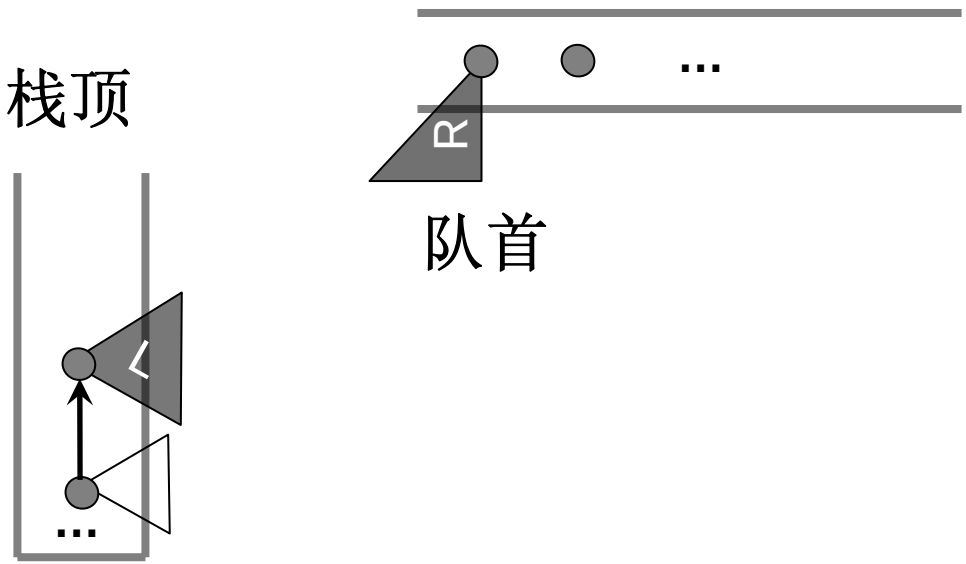


状态和转移

- 在逐趟扫描的过程中，任意时刻的状态是指：当前堆栈和当前队列的格局
- 任意状态下，四种转移动作：left，right，shift和reduce

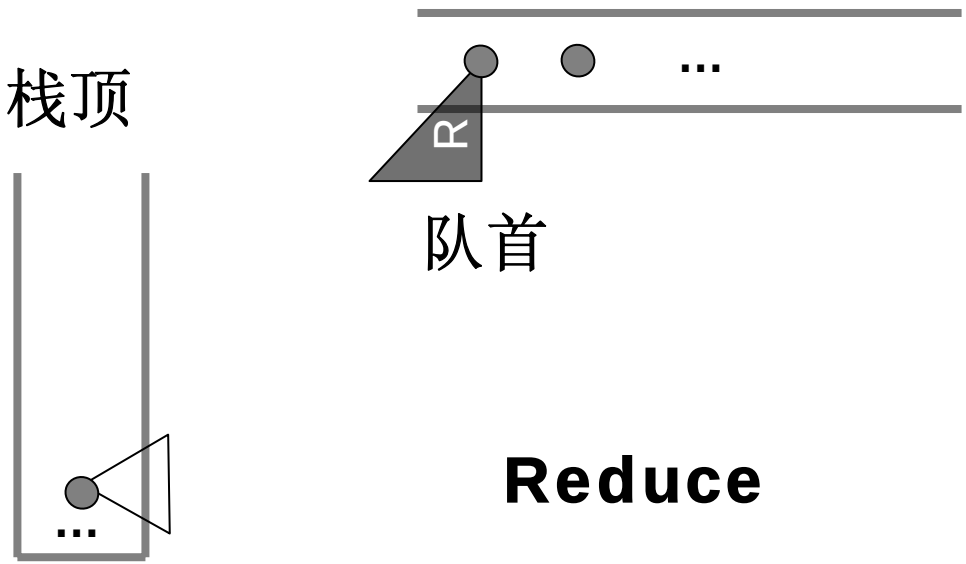
状态和转移

- 在逐趟扫描的过程中，任意时刻的状态是指：当前堆栈和当前队列的格局
- 任意状态下，四种转移动作：left，right，shift和reduce



状态和转移

- 在逐趟扫描的过程中，任意时刻的状态是指：当前堆栈和当前队列的格局
- 任意状态下，四种转移动作：left，right，shift和reduce



状态和转移

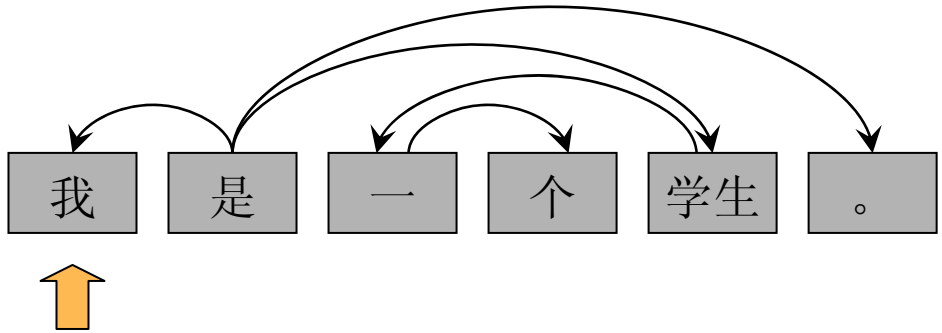
- 动作可以执行的条件
  - Left： 栈顶元素尚无head
  - Right：无
  - Shift：输入队列不空
  - Reduce：栈顶元素有head
- 想一想，为什么？

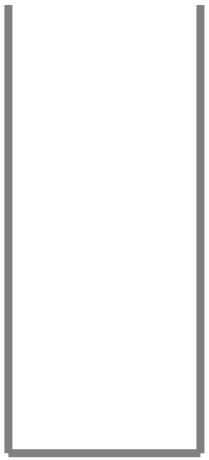
训练和解码

- 训练：
  - 状态转移实例抽取——模拟分析树库中每棵依存树
  - 判别式训练分类器，如SVM
- 解码：
  在单趟扫描的过程中，分类器根据当前栈顶、当前队首以及栈顶和队首所处的上下文，预测需要执行的状态转移操作并转移到新的状态。这一过程持续至输入队列为空

模拟分析

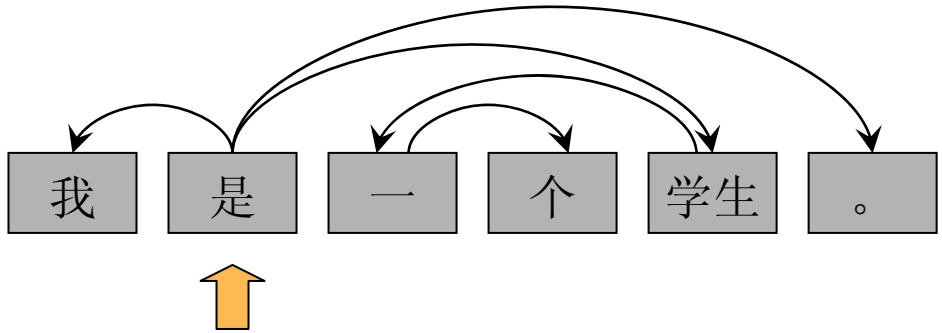
- ((我/PN) 是/VC ((一/CD (个/M)) 学生/NN) (。/PU))

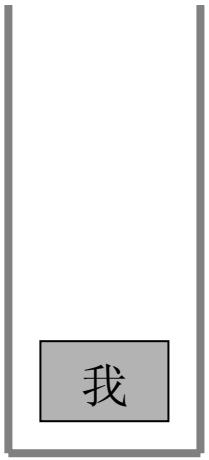




模拟分析

- ((我/PN) 是/VC ((一/CD (个/M)) 学生/NN) (。/PU))

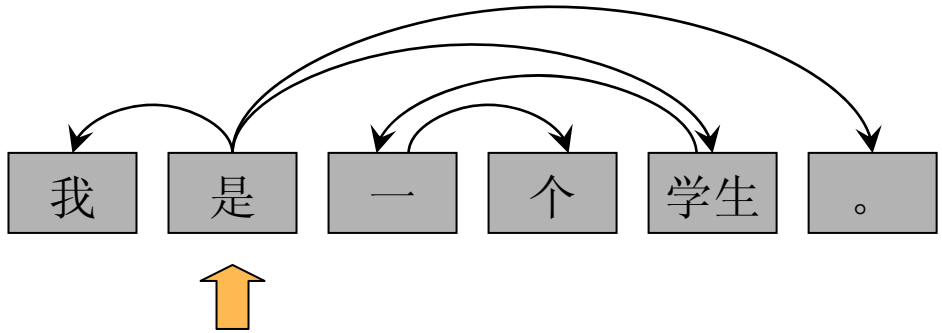


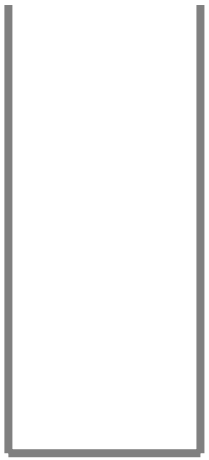


Shift(我)

模拟分析

- ((我/PN) 是/VC ((一/CD (个/M)) 学生/NN) (。/PU))

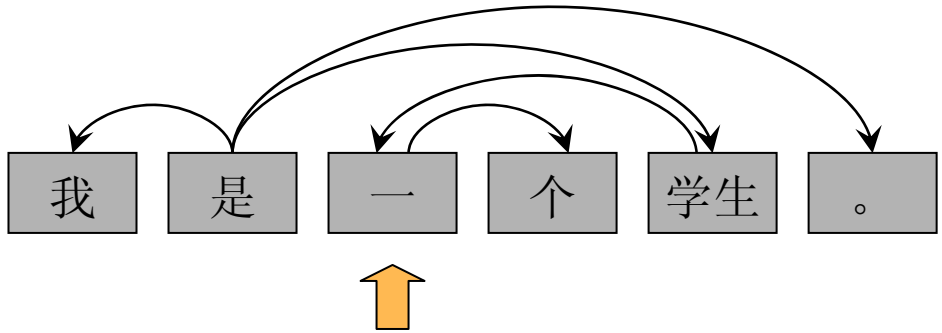


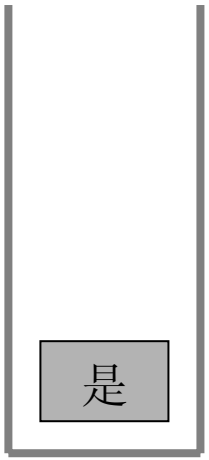


Left(我←是)

模拟分析

- ((我/PN) 是/VC ((一/CD (个/M)) 学生/NN) (。/PU))

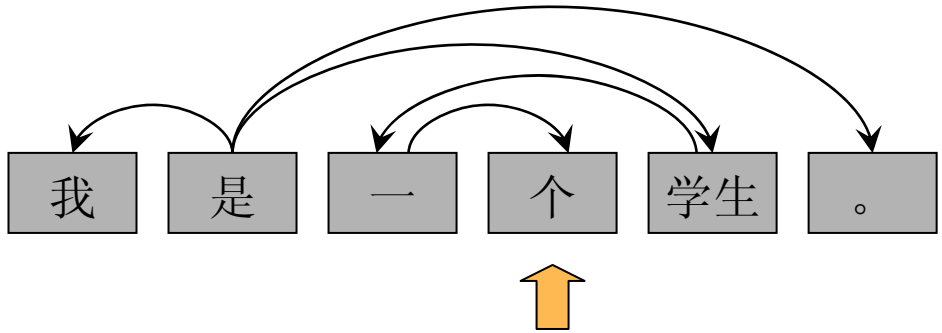


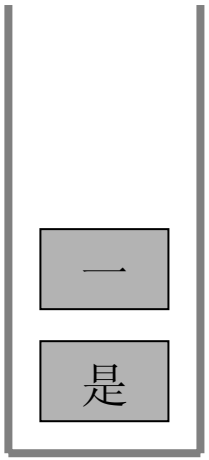


Shift(是)

模拟分析

- ((我/PN) 是/VC ((一/CD (个/M)) 学生/NN) (。/PU))

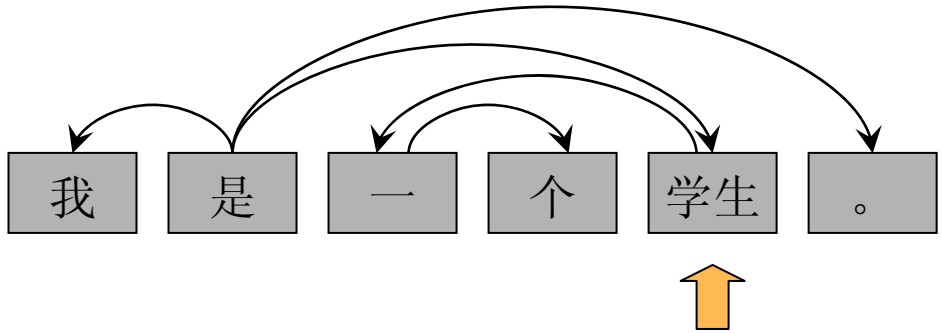


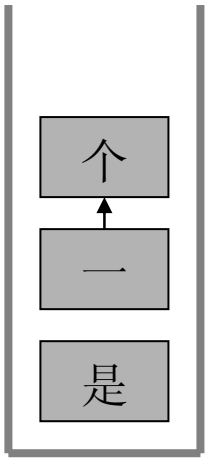


Shift(一)

模拟分析

- ((我/PN) 是/VC ((一/CD (个/M)) 学生/NN) (。/PU))

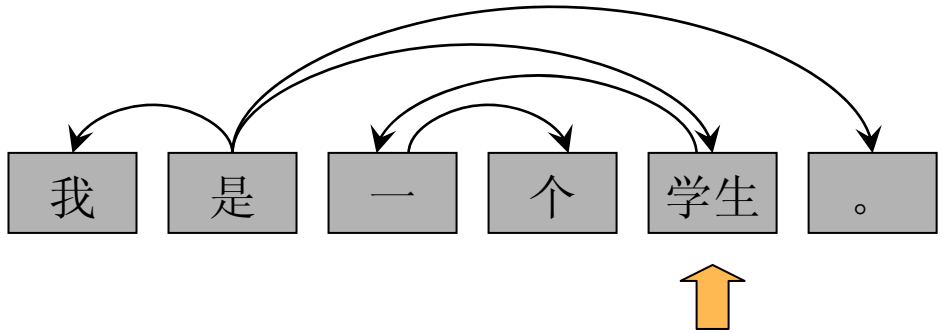


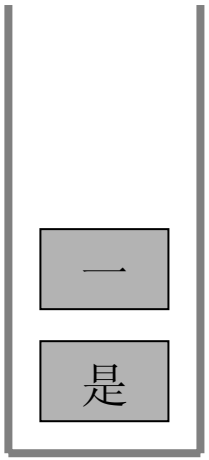


Right(一→个)

模拟分析

- ((我/PN) 是/VC ((一/CD (个/M)) 学生/NN) (。/PU))

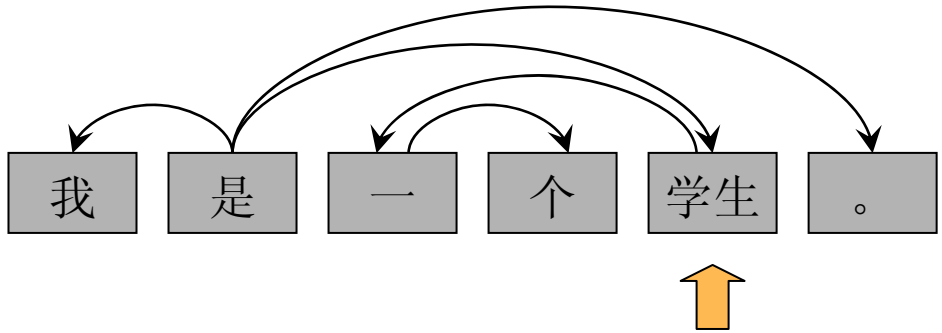


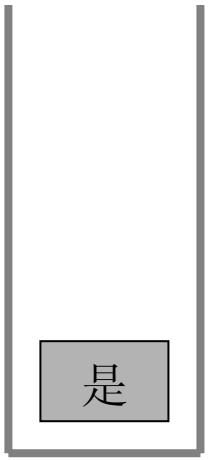


Reduce(个)

模拟分析

- ((我/PN) 是/VC ((一/CD (个/M)) 学生/NN) (。/PU))

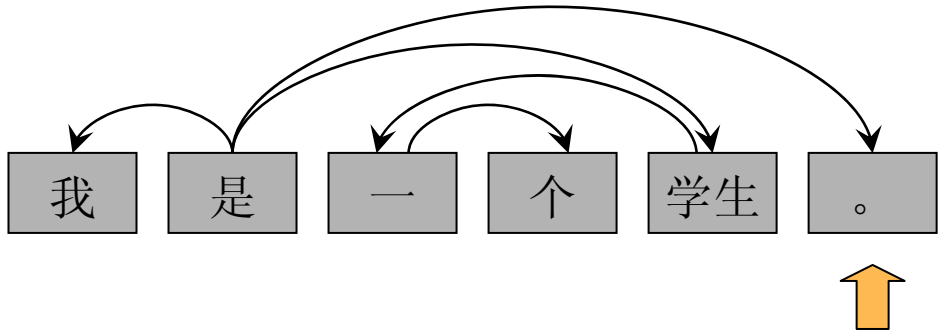


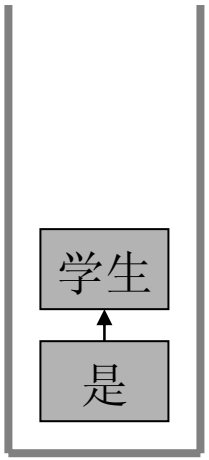


Left(一←学生)

模拟分析

- ((我/PN) 是/VC ((一/CD (个/M)) 学生/NN) (。/PU))

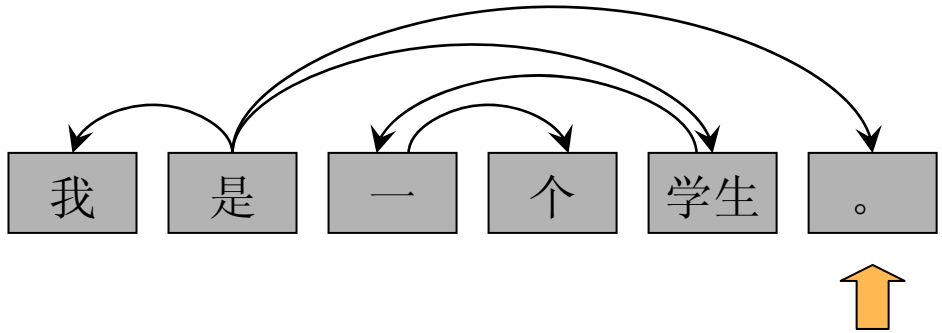


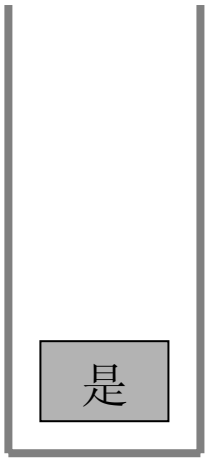


Right(是→学生)

模拟分析

- ((我/PN) 是/VC ((一/CD (个/M)) 学生/NN) (。/PU))

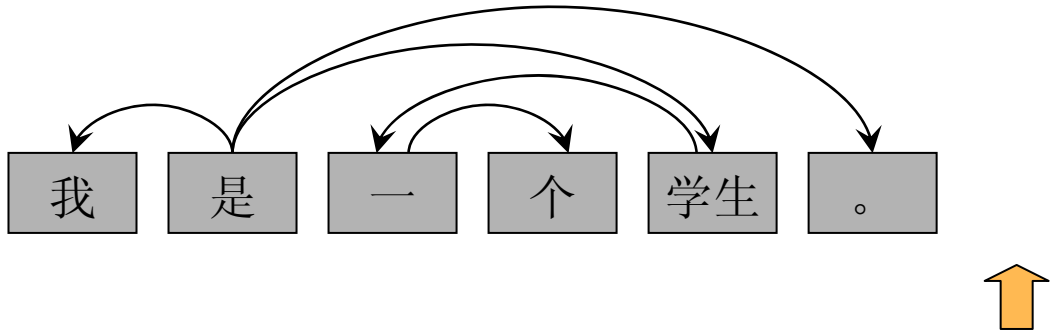


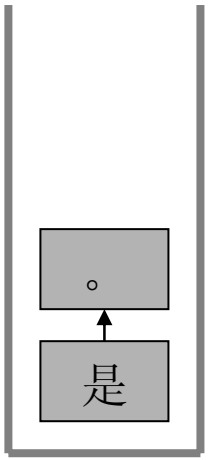


Reduce(学生)

模拟分析

- ((我/PN) 是/VC ((一/CD (个/M)) 学生/NN) (。/PU))

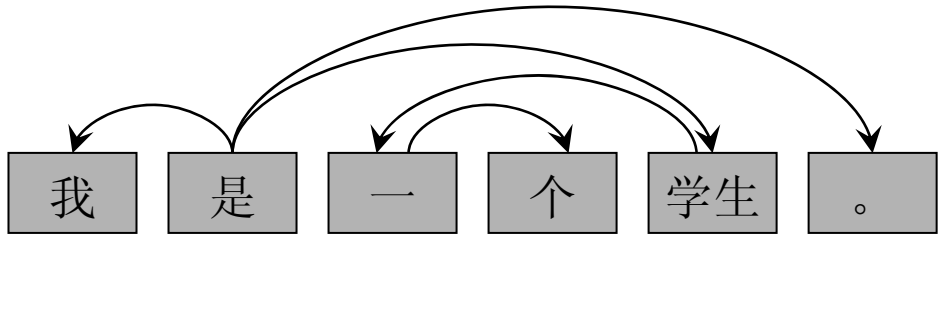


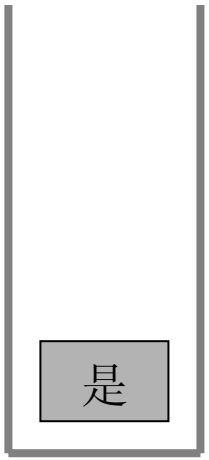


Right(是→。)

模拟分析

- ((我/PN) 是/VC ((一/CD (个/M)) 学生/NN) (。/PU))

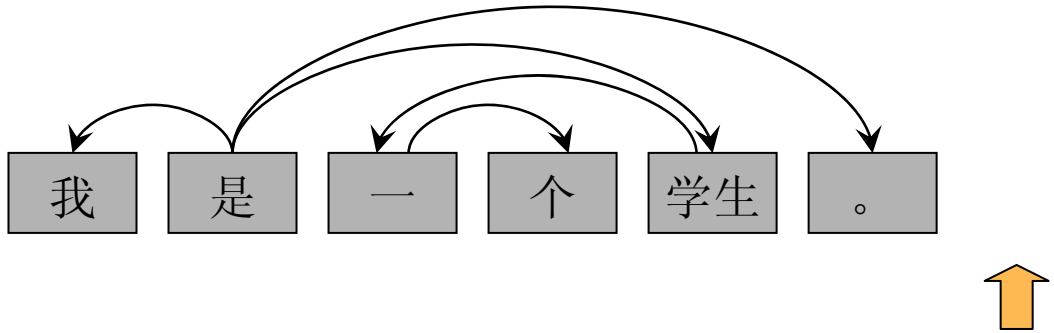


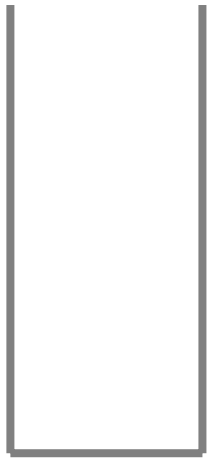


Reduce(。)

模拟分析

- ((我/PN) 是/VC ((一/CD (个/M)) 学生/NN) (。/PU))





Reduce(是)

nbest全局训练

- 单best局部训练，错误传播严重。试考虑：
  - 每步保留N个最好状态
  - 改采用在线全局训练
- nbest全局训练，使得模型预测分数具有比较归一，便于整合其他模型

模型比较

- 生成式依存模型通过简单的极大似然估计即可完成训练，且模型较小。缺点是分析准确率较低
- 最大生成树模型和状态转移模型则需要在训练语料上进行多轮迭代以调节参数，训练耗时长且模型较大。优点是分析准确率高
- 目前流行的是最大生成树模型和状态转移模型中的移进规约模型。其中，最大生成树模型擅于确定远距离的依存关系，而后者则对近距离依存关系识别准确率更高