

2.5 中文分词与条件随机场

林洲汉
上海交大电院

2024年秋季学期

- ▶ **什么是中文分词**
 - ▶ 中文分词的规范
 - ▶ 中文分词中的切分歧义
 - ▶ 中文分词中的未登录词
- ▶ **中文分词的几个主要算法**
 - ▶ 最大匹配法
 - ▶ 最短路径法
 - ▶ 语言模型法
 - ▶ 条件随机场 (CRF)
- ▶ **条件随机场 (CRF) 的几个重要算法**
 - ▶ 给定文本推断分词方案: Viterbi算法
 - ▶ 给定文本计算配分函数: 前向与后向算法
 - ▶ 一些有关CRF的推论 (选讲)
 - ▶ 有监督学习: 最大似然参数估计
- ▶ **中文分词的评价指标**

- ▶ **什么是中文分词**
 - ▶ 中文分词的规范
 - ▶ 中文分词中的切分歧义
 - ▶ 中文分词中的未登录词
- ▶ **中文分词的几个主要算法**
 - ▶ 最大匹配法
 - ▶ 最短路径法
 - ▶ 语言模型法
 - ▶ 条件随机场 (CRF)
- ▶ **条件随机场 (CRF) 的几个重要算法**
 - ▶ 给定文本推断分词方案: Viterbi算法
 - ▶ 给定文本计算配分函数: 前向与后向算法
 - ▶ 一些有关CRF的推论 (选讲)
 - ▶ 有监督学习: 最大似然参数估计
- ▶ **中文分词的评价指标**

中文分词的重要性



南京市/长江/大桥



南京/市长/江大桥

- 自动分词是汉语句子分析的基础
- 分词具有广泛的应用
 - 词频统计
 - 词典编撰
 - 主题分类
- 分词直接影响了后续文本处理算法所用的特征。
 - 词包法 (bag-of-words) 的文本分类
 - 句法解析
 -

- 中文分词规范问题（《信息处理用限定汉语分词规范（GB13715）》）
 - - 汉语中什么是词？两个不清的界限：
 - (1) 单字词与词素：
 - 新华社25日**讯**
 - (2) 词与短语：
 - 花草，湖边，房顶，鸭蛋，小鸟，担水，
一层

合并原则

1. 语义上无法由组合成分直接相加而得到的字串应该合并为一个分词单位。
2. 语类无法由组合成分直接得到的字串应该合并为一个分词单位。
3. 附着性语(词)素和前后词合并为一个分词单位。
4. 使用频率高或共现率高的字串尽量合并为一个分词单位。
5. 双音节加单音节的偏正式名词尽量合并为一个分词单位。
6. 双音节结构的偏正式动词应尽量合并为一个分词单位。

切分原则

1. 有明显分隔符标记的应该切分。
2. 内部结构复杂、合并起来过于冗长的词尽量切分。

中文分词中的切分歧义

中国人/为了/实现/自己/的/梦想

中/国人/为了/实现/自己/的/梦想

中国/人为/了/实现/自己/的/梦想

中国人为了自己的梦想

中文分词中的切分歧义

中 国 人 / 为 了 / 实 现 / 自 己 / 的 / 梦 想

中 / 国 人 / 为 了 / 实 现 / 自 己 / 的 / 梦 想

中 国 / 人 为 / 了 / 实 现 / 自 己 / 的 / 梦 想

中 国 人 为 了 实 现 自 己 的 梦 想

中文分词中的切分歧义：交集型歧义

结 合 成 分 子

为 人 民 工 作

中 国 产 品 质 量

部 分 居 民 生 活 水 平

中文分词中的切分歧义：组合型歧义

学 生 会 / 来 / 找 / 你
学 生 会 / 来 / 找 / 你

门 把 手 / 弄 / 坏 / 了
门 把 手 / 弄 / 坏 / 了
门 把 手 / 弄 / 坏 / 了

中文分词中的切分歧义：经验数字

梁南元（1987）曾经对一个含有48,092字的自然科学、社会科学样本进行了统计，结果交集型切分歧义有518个，组合型切分歧义有42个。据此推断：

1. 中文文本中切分歧义的出现频度约为1.2次/100字
2. 交集型切分歧义与组合型切分歧义的出现比例约为12:1。

中文分词中未登录词的识别

未登录词，即unknown words，简记作UNK。代表模型在训练阶段没有见过的单词。

1. 人名、地名、组织机构名等，例如：

盛中国，张建国，蔡国庆，蔡英文，水皮，高升，高山，夏天，温馨，武夷山，平川三太郎，约翰·斯特朗，詹姆斯·埃尔德

2. 新出现的词汇、术语、个别俗语等，例如：

抖音，新冠，奥利给，楼歪歪

中文分词中未登录词的识别

1. 他还兼任任何应钦在福州办的东路军军官学校的政治教官。
2. 大不列颠及北爱尔兰联合王国外交和英联邦事务大臣、议会议员杰克·斯特劳阁下在联合国安理会就伊拉克问题发言。
3. 坐落于江苏省南京市玄武湖公园内的夏璞墩是晋代著名的文学家、科学家夏璞的衣冠冢。

中文分词中未登录词的识别

错误类型			错误数	比例 (%)			例子
集外词	命名实体	人名	31	25.83	55.0	98.33	约翰 斯坦贝克
		地名	11	9.17			米苏拉塔
		组织机构名	10	8.33			泰党
		时间和数字	14	11.67			37万兆
	专业术语		4	3.33	脱氧核糖核酸		
	普通生词		48	40.00	致病原		
	切分歧义			2	1.67		歌名为
合计			120	100			

从互联网上随机摘取了418个句子，共含11,739个词， 19,777个汉字（平均每个句长约为28个词，**每个词约含 1.68个汉字**）。

- ▶ **什么是中文分词**
 - ▶ 中文分词的规范
 - ▶ 中文分词中的切分歧义
 - ▶ 中文分词中的未登录词
- ▶ **中文分词的几个主要算法**
 - ▶ 最大匹配法
 - ▶ 最短路径法
 - ▶ 语言模型法
 - ▶ 条件随机场 (CRF)
- ▶ **条件随机场 (CRF) 的几个重要算法**
 - ▶ 给定文本推断分词方案: Viterbi算法
 - ▶ 给定文本计算配分函数: 前向与后向算法
 - ▶ 一些有关CRF的推论 (选讲)
 - ▶ 有监督学习: 最大似然参数估计
- ▶ **中文分词的评价指标**

中文分词的基本算法



中文分词的基本算法



中文分词的基本算法：最大匹配法

最大匹配算法 (Maximum Matching)

正向最大匹配算法 (Forward MM, FMM)

逆向最大匹配算法 (Backward MM, BMM)

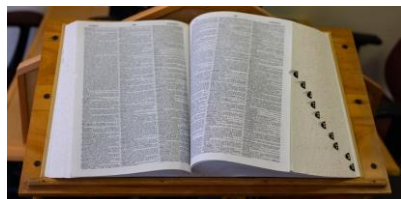
中文分词的基本算法：最大匹配法

最大匹配算法 (Maximum Matching)

正向最大匹配算法 (Forward MM, FMM)

逆向最大匹配算法 (Backward MM, BMM)

他 是 研 究 生 物 化 学 的



词典:

最长单词: 6字

最短单词: 1字

中文分词的基本算法：最大匹配法

最大匹配算法 (Maximum Matching)

正向最大匹配算法 (Forward MM, FMM)

逆向最大匹配算法 (Backward MM, BMM)

他 / 是 研 究 生 物 化 学 的



词典:

最长单词: 6字

最短单词: 1字



中文分词的基本算法：最大匹配法

最大匹配算法 (Maximum Matching)

正向最大匹配算法 (Forward MM, FMM)

逆向最大匹配算法 (Backward MM, BMM)

他 / 是 / 研 究 生 物 化 学 的



词典:

最长单词：6字

最短单词：1字



中文分词的基本算法：最大匹配法

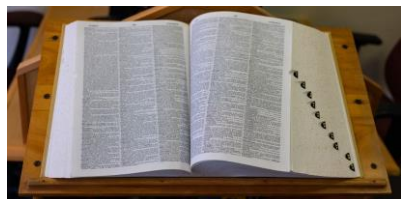
最大匹配算法 (Maximum Matching)

正向最大匹配算法 (Forward MM, FMM)

逆向最大匹配算法 (Backward MM, BMM)

他 / 是 / 研 究 生 / 物 化 学 的

⋮



词典:

最长单词: 6字

最短单词: 1字

中文分词的基本算法：最大匹配法

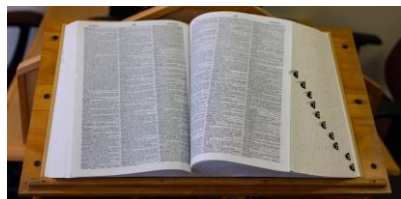
最大匹配算法 (Maximum Matching)

正向最大匹配算法 (Forward MM, FMM)

逆向最大匹配算法 (Backward MM, BMM)

他 / 是 / 研 究 生 / 物 化 / 学 的

⋮



词典:

最长单词: 6字

最短单词: 1字

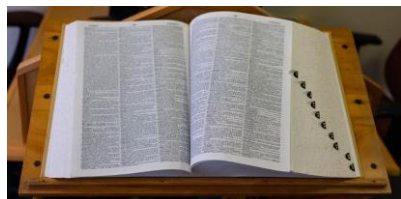
中文分词的基本算法：最大匹配法

最大匹配算法 (Maximum Matching)

正向最大匹配算法 (Forward MM, FMM)

逆向最大匹配算法 (Backward MM, BMM)

他 / 是 / 研 究 生 / 物 化 / 学 的



词典:

最长单词：6字

最短单词：1字

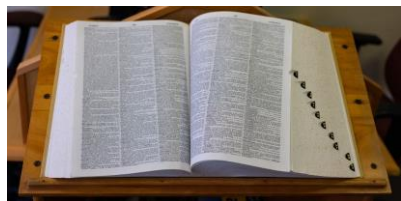
中文分词的基本算法：最大匹配法

最大匹配算法 (Maximum Matching)

正向最大匹配算法 (Forward MM, FMM)

逆向最大匹配算法 (Backward MM, BMM)

他 / 是 / 研 究 生 / 物 化 / 学 / 的



词典:

最长单词: 6字

最短单词: 1字

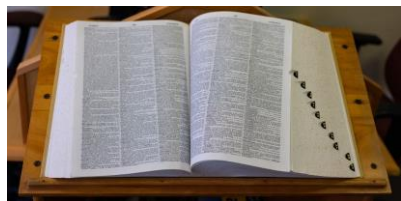
中文分词的基本算法：最大匹配法

最大匹配算法 (Maximum Matching)

正向最大匹配算法 (Forward MM, FMM)

逆向最大匹配算法 (Backward MM, BMM)

他 是 研 究 生 物 化 学 的



词典:

最长单词：6字

最短单词：1字

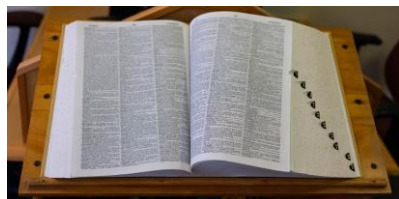
中文分词的基本算法：最大匹配法

最大匹配算法 (Maximum Matching)

正向最大匹配算法 (Forward MM, FMM)

逆向最大匹配算法 (Backward MM, BMM)

他 是 研 究 生 物 化 学 / 的



词典:

最长单词: 6字

最短单词: 1字



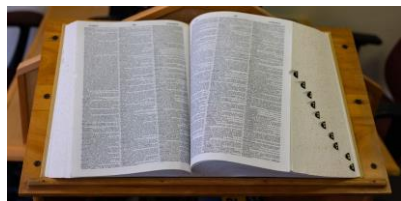
中文分词的基本算法：最大匹配法

最大匹配算法 (Maximum Matching)

正向最大匹配算法 (Forward MM, FMM)

逆向最大匹配算法 (Backward MM, BMM)

他 是 研 究 生 物 / 化 学 / 的



词典:

最长单词: 6字

最短单词: 1字

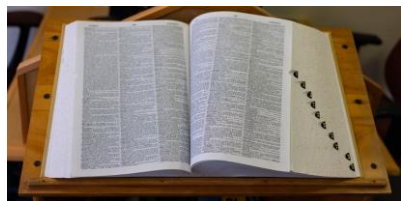
中文分词的基本算法：最大匹配法

最大匹配算法 (Maximum Matching)

正向最大匹配算法 (Forward MM, FMM)

逆向最大匹配算法 (Backward MM, BMM)

他 是 研 究 / 生 物 / 化 学 / 的



词典:

最长单词: 6字

最短单词: 1字

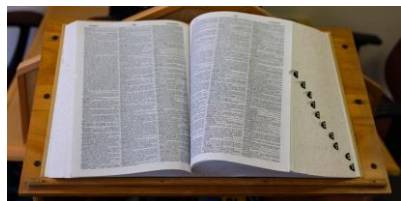
中文分词的基本算法：最大匹配法

最大匹配算法 (Maximum Matching)

正向最大匹配算法 (Forward MM, FMM)

逆向最大匹配算法 (Backward MM, BMM)

他 是 / 研 究 / 生 物 / 化 学 / 的



词典:

最长单词：6字

最短单词：1字

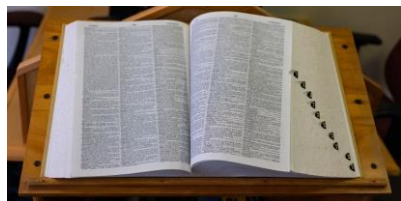
中文分词的基本算法：最大匹配法

最大匹配算法 (Maximum Matching)

正向最大匹配算法 (Forward MM, FMM)

逆向最大匹配算法 (Backward MM, BMM)

他 / 是 / 研 究 / 生 物 / 化 学 / 的



词典:

最长单词: 6字

最短单词: 1字

中文分词的基本算法：最大匹配法

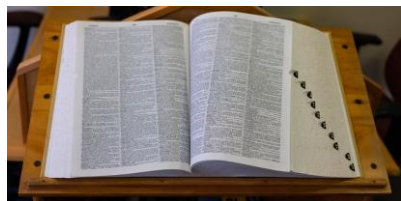
最大匹配算法 (Maximum Matching)

正向最大匹配算法 (Forward MM, FMM)

逆向最大匹配算法 (Backward MM, BMM)

逆向最大匹配算法结果：

他 / 是 / 研 究 / 生 物 / 化 学 / 的



正向最大匹配算法结果：

他 / 是 / 研 究 生 / 物 / 化 学 / 的

词典：

最长单词：6字

最短单词：1字

中文分词的基本算法：最大匹配法

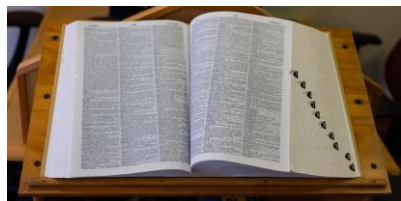
最大匹配算法 (Maximum Matching)

正向最大匹配算法 (Forward MM, FMM)

逆向最大匹配算法 (Backward MM, BMM)

优点：

- 程序简单易行，开发周期短
- 仅需要很少的语言资源（词表），不需要任何词法、句法、语义资源



词典：

最长单词：6字

最短单词：1字

缺点：

- 歧义消解的能力差

中文分词的基本算法



中文分词的基本算法：最短路径法

中→国→人→为→了→实→现→自→己→的→梦→想



从以上所有路径中，选择路径最短的
(词数最少的)作为最终分词结果



中 国 人 / 为 了 / 实 现 / 自 己 / 的 / 梦 想

中文分词的基本算法：最短路径法

中→国→人→为→了→实→现→自→己→的→梦→想



优点：

- 切分原则明确，符合汉语自身规律
- 需要的语言资源（词表）也不多。

缺点：

- 对许多歧义字段难以区分，最短路径有多条时，选择最终的输出结果缺乏应有的标准
- 字串长度较大和选取的最短路径数增大时，长度相同的路径数急剧增加，选择最终正确的结果困难越来越大

中文分词的基本算法



中文分词的基本算法：语言模型法

设句子 $S = \{s_1, s_2, \dots, s_N\}$ 可切分为 K 个单词 $W = \{w_1, w_2, \dots, w_K\}$, 则此任务为找到最优的 W , 即 W^* :

$$\begin{aligned} W^* &= \operatorname{argmax}_W P(W|S) \\ &= \operatorname{argmax}_W P(W) \boxed{P(S|W)} \\ &= \operatorname{argmax}_W P(W) \quad = 1 \end{aligned}$$

对于给定的 $P(W)$, 其概率由预先训练好的语言模型给出。

中文分词的基本算法：语言模型法

计算 $\operatorname{argmax}_W(\cdot)$ 的方法：

$$W^* = \operatorname{argmax}_W P(W|S)$$

穷举所有可能的 W ，分别计算其对应的 $P(W)$ ，挑出最大的即为 W^* 。

他 是 研 究 生 物 化 学 的

他 / 是 / 研 究 / 生 物 / 化 学 / 的 ✓

他 / 是 / 研 究 生 / 物 / 化 学 / 的 ✗

他 / 是 / 研 究 生 / 物 化 / 学 / 的 ✗

中文分词的基本算法：语言模型法

计算 $\operatorname{argmax}_W(\cdot)$ 的方法：

$$W^* = \operatorname{argmax}_W P(W|S)$$

束搜索 (beam search)：当穷举所有可能的 W 变得极多而无法一一计算时，利用束搜索从左至右地确定分词方案 W^* 。

他 是 研 究 生 物 化 学 的

他 / 是

他 / 是 研

他 / 是 研 究

中文分词的基本算法：语言模型法

计算 $\operatorname{argmax}_W(\cdot)$ 的方法：

$$W^* = \operatorname{argmax}_W P(W|S)$$

束搜索 (beam search)：当穷举所有可能的 W 变得极多而无法一一计算时，利用束搜索从左至右地确定分词方案 W^* 。

他 是 研 究 生 物 化 学 的

	他 / 是 / 研	×
	他 / 是 / 研 究	√
他 / 是	他 / 是 / 研 究 生	√
	他 / 是 研 / 究	×
他 / 是 研	他 / 是 研 / 究 生	×
	他 / 是 研 / 究 生 物	×
他 / 是 研 究	他 / 是 研 究 / 生	×
	他 / 是 研 究 / 生 物	√
	他 / 是 研 究 / 生 物 化	×

中文分词的基本算法：语言模型法

计算 $\operatorname{argmax}_W(\cdot)$ 的方法：

$$W^* = \operatorname{argmax}_W P(W|S)$$

束搜索（beam search）：当穷举所有可能的 W 变得极多而无法一一计算时，利用束搜索从左至右地确定分词方案 W^* 。

他是研究生物化学的

	他 / 是 / 研 究 / 生	×
他 / 是 / 研 究	他 / 是 / 研 究 / 生 物	√
	他 / 是 / 研 究 / 生 物 化	√
	他 / 是 / 研 究 生 / 物	×
他 / 是 / 研 究 生	他 / 是 / 研 究 生 / 物 化	√
	他 / 是 / 研 究 生 / 物 化 学	×
	他 / 是 研 究 / 生 物 / 化	×
他 / 是 研 究 / 生 物	他 / 是 研 究 / 生 物 / 化 学	×
	他 / 是 研 究 / 生 物 / 化 学 的	×

中文分词的基本算法：语言模型法

计算 $\operatorname{argmax}_W(\cdot)$ 的方法：

$$W^* = \operatorname{argmax}_W P(W|S)$$

束搜索（beam search）：当穷举所有可能的 W 变得极多而无法一一计算时，利用束搜索从左至右地确定分词方案 W^* 。

他是研究生物化学的

	他 / 是 / 研 究 / 生 物 / 化	×
他 / 是 / 研 究 / 生 物	他 / 是 / 研 究 / 生 物 / 化 学	√
	他 / 是 / 研 究 / 生 物 / 化 学 的	√
	他 / 是 / 研 究 / 生 物 化 / 学	√
他 / 是 / 研 究 / 生 物 化	他 / 是 / 研 究 / 生 物 化 / 学 的	×
	∅	×
	他 / 是 / 研 究 生 / 物 化 / 学	×
他 / 是 / 研 究 生 / 物 化	他 / 是 / 研 究 生 / 物 化 / 学 的	×
	∅	×

中文分词的基本算法：语言模型法

计算 $\operatorname{argmax}_W(\cdot)$ 的方法：

$$W^* = \operatorname{argmax}_W P(W|S)$$

束搜索 (beam search)：当穷举所有可能的 W 变得极多而无法一一计算时，利用束搜索从左至右地确定分词方案 W^* 。

他是研究生物化学的

他 / 是 / 研 究 / 生 物 / 化 学

他 / 是 / 研 究 / 生 物 / 化 学 / 的 ✓

∅ ×

∅ ×

∅ ×

他 / 是 / 研 究 / 生 物 / 化 学 的

∅ ×

∅ ×

他 / 是 / 研 究 / 生 物 化 / 学

他 / 是 / 研 究 / 生 物 化 / 学 / 的 ✓

∅ ×

∅ ×

✓

中文分词的基本算法：语言模型法

计算 $\operatorname{argmax}_W(\cdot)$ 的方法：

$$W^* = \operatorname{argmax}_W P(W|S)$$

束搜索 (beam search)：当穷举所有可能的 W 变得极多而无法一一计算时，利用束搜索从左至右地确定分词方案 W^* 。

无法保证找到全局最优解，但是可以在可接受的计算量下达到可接受的次优解

中文分词的基本算法：语言模型法

$$W^* = \operatorname{argmax}_W P(W|S)$$

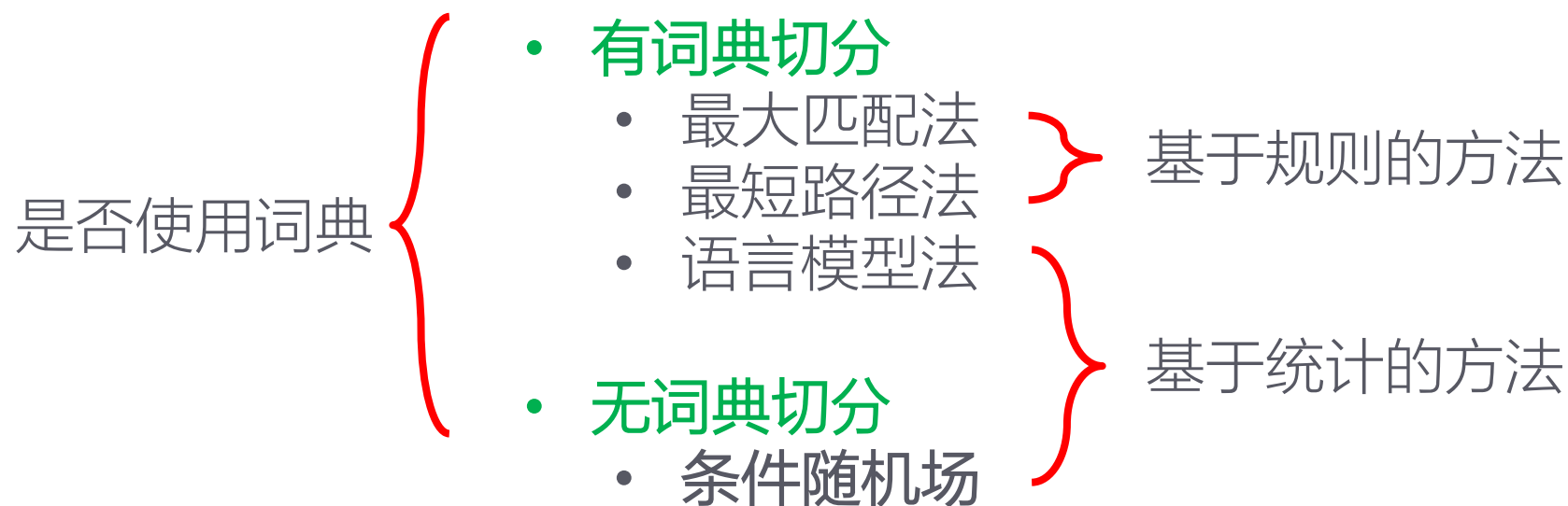
优点：

- 减少了很多手工标注的工作
- 在训练语料规模足够大和覆盖领域足够多时，可以获得较高的切分正确率

缺点：

- 训练语料的规模和覆盖领域不好把握
- 计算量较大

中文分词的基本算法



中文分词的基本算法：条件随机场的问题设定

将分词看作是序列标注问题，依次对序列中每个字赋予一个标签，再根据标签合并同属于一个单词的字符：

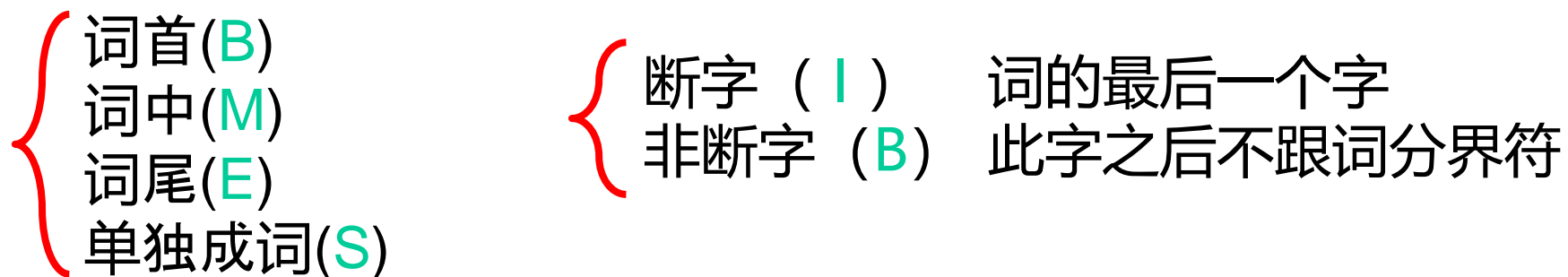
{ 词首(B)
词中(M)
词尾(E)
单独成词(S)

中 国 人 / 为 了 / 实 现 / 自 己 / 的 / 梦 想

B	✓			✓		✓		✓		✓	
M		✓									
E			✓		✓		✓		✓		✓
S									✓		
	中	国	人	为	了	实	现	自	己	的	梦 想

中文分词的基本算法：条件随机场的问题设定

将分词看作是序列标注问题，依次对序列中每个字赋予一个标签，再根据标签合并同属于一个单词的字符：



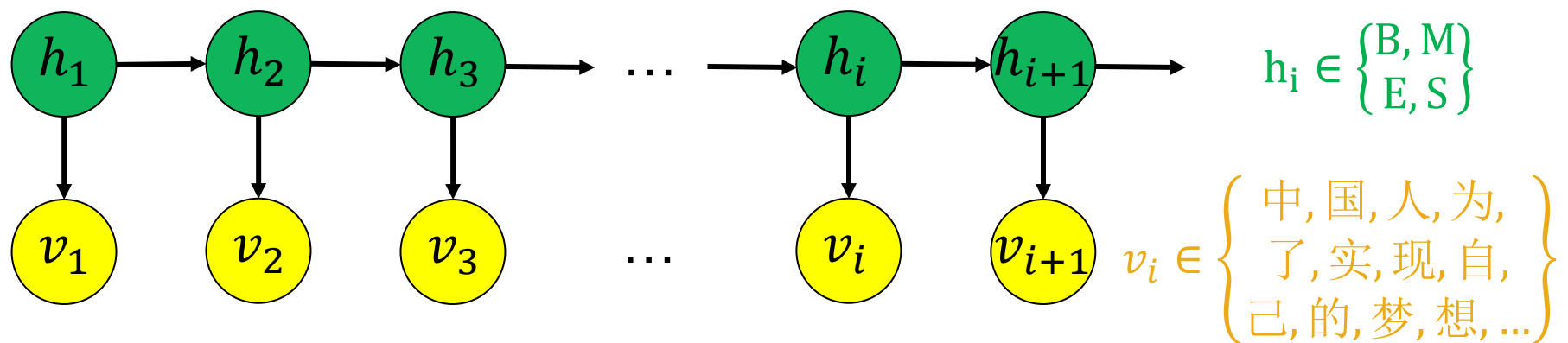
中 国 人 / 为 了 / 实 现 / 自 己 / 的 / 梦 想

B ✓ ✓ ✓ ✓ ✓ ✓

I ✓ ✓ ✓ ✓ ✓

中 国 人 为 了 实 现 自 己 的 梦 想

中文分词的基本算法：乱入的HMM



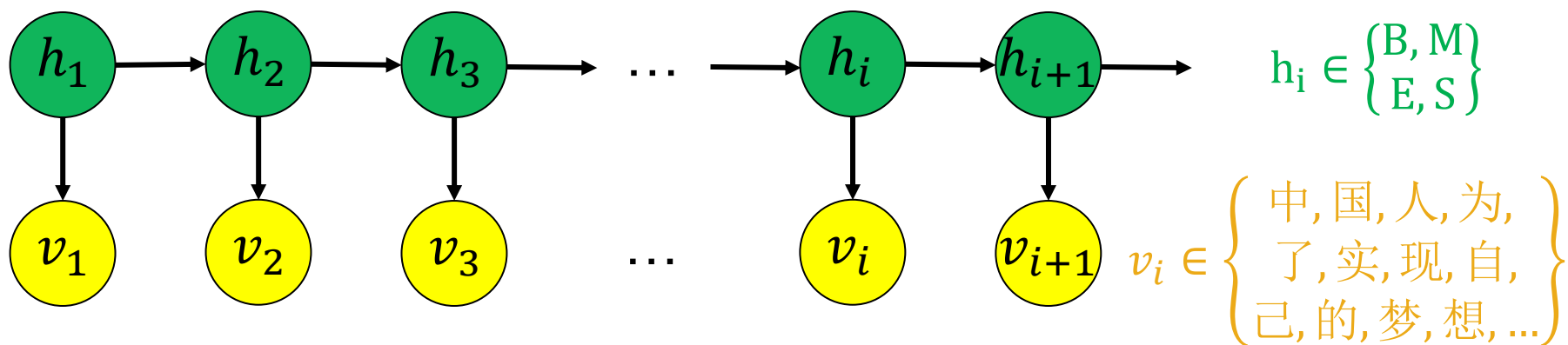
中 国 人 / 为 了 / 实 现 / 自 己 / 的 / 梦 想

B ✓ ✓ ✓ ✓ ✓ ✓

I ✓ ✓ ✓ ✓ ✓ ✓

中 国 人 为 了 实 现 自 己 的 梦 想

中文分词的基本算法：乱入的HMM

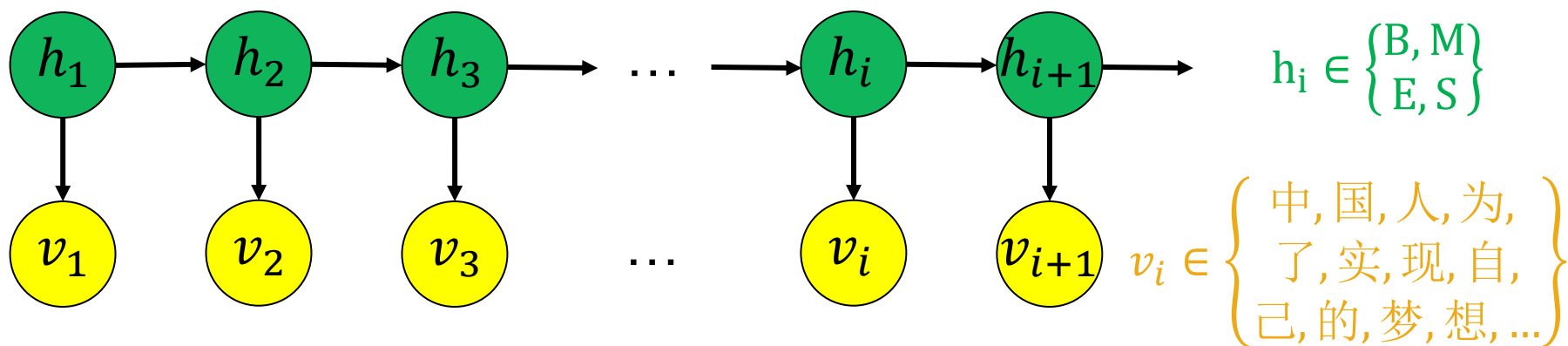


$$\operatorname{argmax}_{\underline{h}} P(\underline{h} | \underline{v}) = \operatorname{argmax}_{\underline{h}} P(\underline{h}, \underline{v})$$

$$= \operatorname{argmax}_{\underline{h}} (P(\underline{v} | \underline{h}) \cdot P(\underline{h}))$$

$$= \operatorname{argmax}_{\underline{h}} \left(\prod_{j=1}^N P(v_j | h_j) P(h_1) \prod_{i=1}^{N-1} P(h_{i+1} | h_i) \right)$$

中文分词的基本算法：乱入的HMM



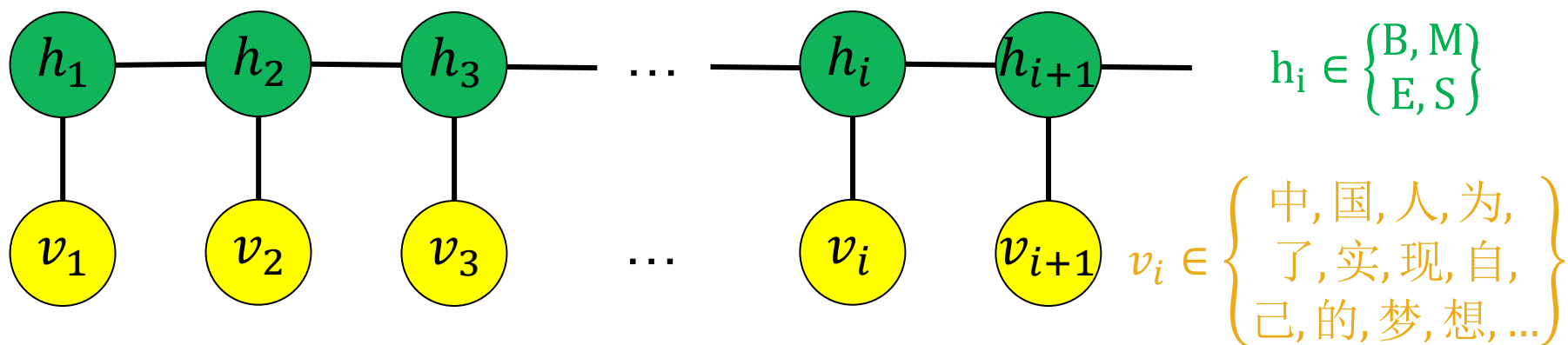
$$\operatorname{argmax}_{\underline{h}} \underline{P(\underline{h} | \underline{v})} = \operatorname{argmax}_{\underline{h}} \underline{P(\underline{h} | \underline{v})}$$

$$= \operatorname{argmax}_{\underline{h}} \left(\underline{P(\underline{h} | \underline{v})} \cdot P(\underline{h}) \right)$$

$$= \operatorname{argmax}_{\underline{h}} \left(\prod_{j=1}^N P(\underline{h}_j | \underline{v}_j) P(\underline{h}_j) \prod_{i=1}^{N-1} P(\underline{h}_{i+1} | \underline{h}_i) \right)$$

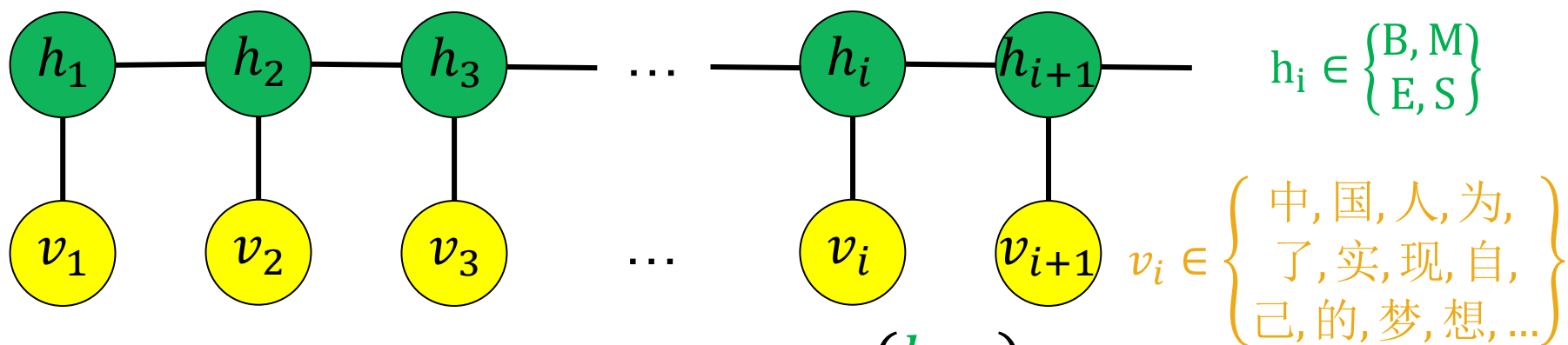
不通过贝叶斯公式，
而是直接对条件概率
简单直接地进行拟合。

中文分词的基本算法：CRF概念



$$\operatorname{argmax}_{\underset{h}{\text{green}}} P(\underset{h}{\text{green}} | \underset{v}{\text{orange}}) = \operatorname{argmax}_{\underset{h}{\text{green}}} (\operatorname{score}(\underset{h}{\text{green}}, \underset{v}{\text{orange}}) \geq 0)$$

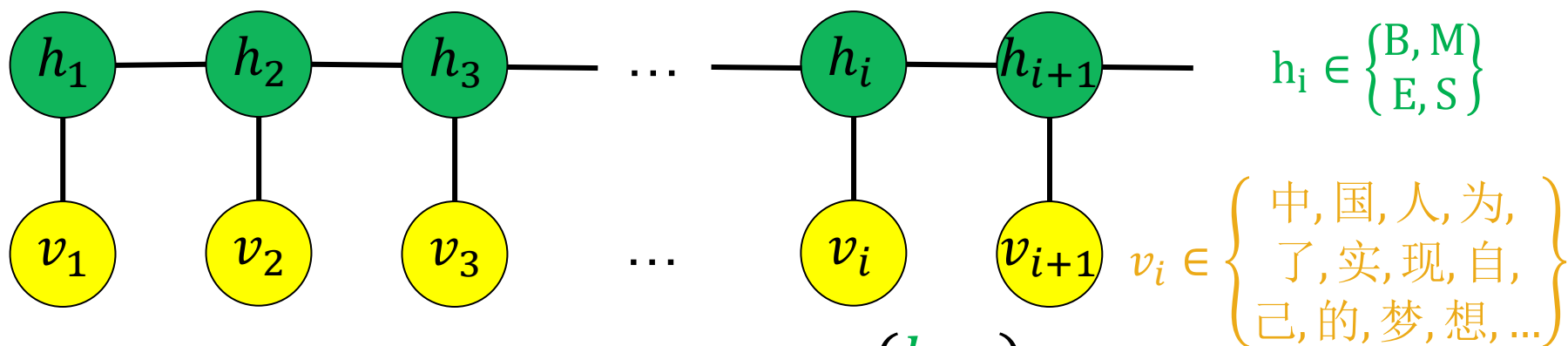
中文分词的基本算法：CRF概念



$$\operatorname{argmax}_{\underline{h}} P(\underline{h}|\underline{v}) = \operatorname{argmax}_{\underline{h}} \frac{\operatorname{score}(\underline{h}, \underline{v})}{\sum_{\underline{h}} \operatorname{score}(\underline{h}, \underline{v})} \quad (\operatorname{score}(\underline{h}, \underline{v}) \geq 0)$$

$$\operatorname{score}(\underline{h}, \underline{v}) = F_k(\underline{h}, \underline{v})$$

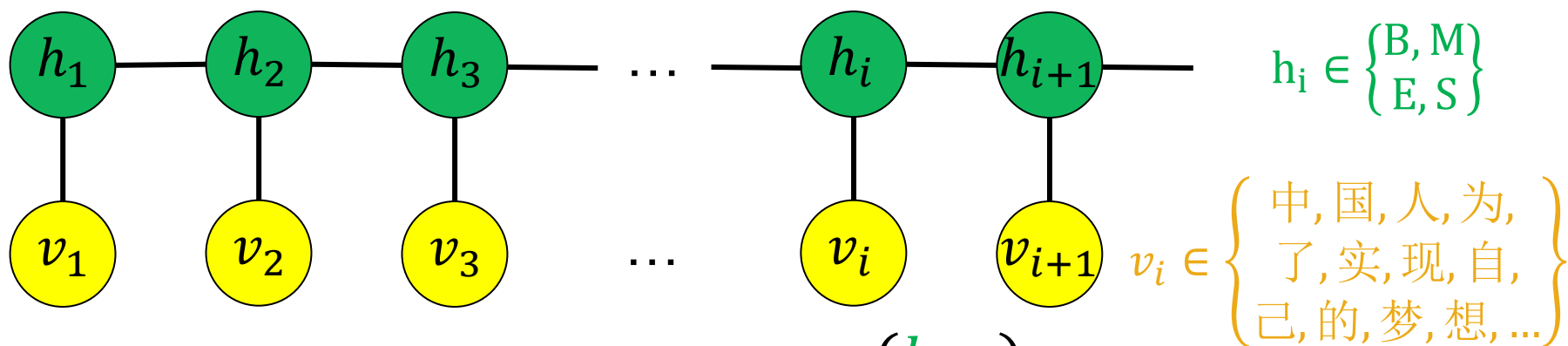
中文分词的基本算法：CRF概念



$$\operatorname{argmax}_{\underline{h}} P(\underline{h} | \underline{v}) = \operatorname{argmax}_{\underline{h}} \frac{\operatorname{score}(\underline{h}, \underline{v})}{\sum_{\underline{h}} \operatorname{score}(\underline{h}, \underline{v})} \quad (\operatorname{score}(\underline{h}, \underline{v}) \geq 0)$$

$$\operatorname{score}(\underline{h}, \underline{v}) = \sum_{k=1}^K w_k F_k(\underline{h}, \underline{v})$$

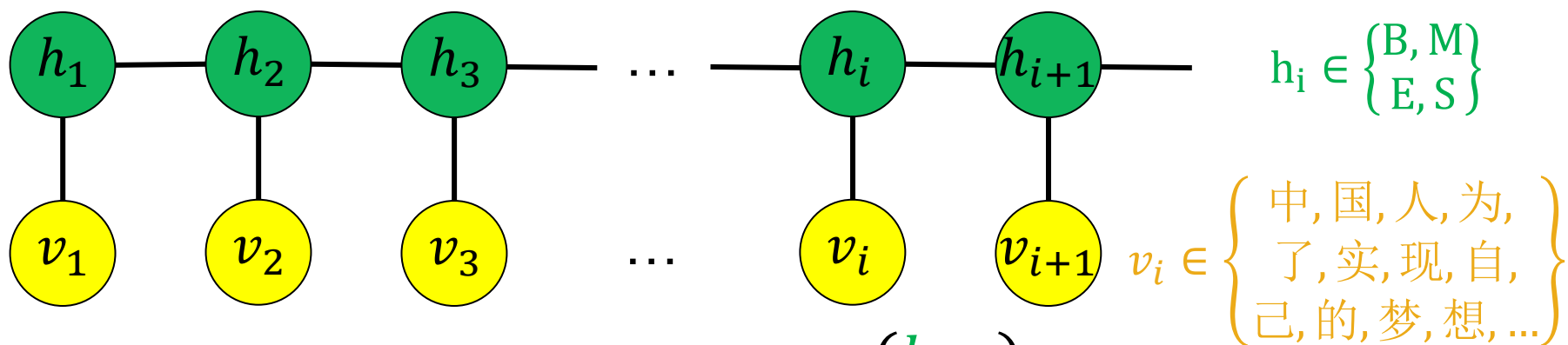
中文分词的基本算法：CRF概念



$$\operatorname{argmax}_{\underline{h}} P(\underline{h} | \underline{v}) = \operatorname{argmax}_{\underline{h}} \frac{\operatorname{score}(\underline{h}, \underline{v})}{\sum_{\underline{h}} \operatorname{score}(\underline{h}, \underline{v})} \quad (\operatorname{score}(\underline{h}, \underline{v}) \geq 0)$$

$$\operatorname{score}(\underline{h}, \underline{v}) = \exp \left(\sum_{k=1}^K w_k F_k(\underline{h}, \underline{v}) \right)$$

中文分词的基本算法：CRF概念

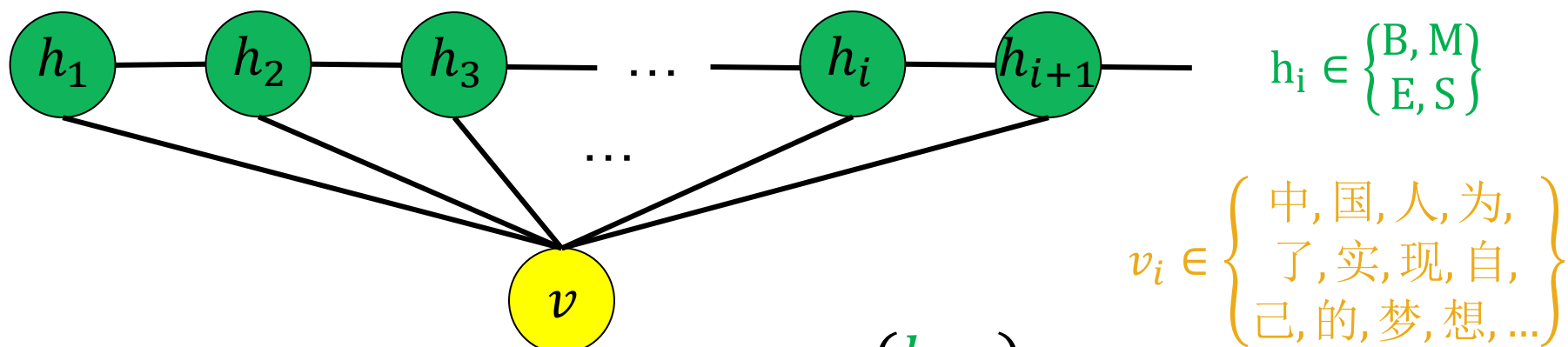


$$\operatorname{argmax}_{\mathbf{h}} P(\mathbf{h}|\mathbf{v}) = \operatorname{argmax}_{\mathbf{h}} \frac{\operatorname{score}(\mathbf{h}, \mathbf{v})}{\sum_{\mathbf{h}} \operatorname{score}(\mathbf{h}, \mathbf{v})} \quad (\operatorname{score}(\mathbf{h}, \mathbf{v}) \geq 0)$$

$$\operatorname{score}(\mathbf{h}, \mathbf{v}) = \exp \left(\sum_{k=1}^K w_k F_k(\mathbf{h}, \mathbf{v}) \right)$$

$$\begin{aligned} F_k(\mathbf{h}, \mathbf{v}) &= F_k(h_1, h_2, \dots, h_i, \dots, h_N, v_1, v_2, \dots, v_i, \dots, v_N) \\ &= \sum_{i=2}^N f_k(h_i, h_{i-1}, v, i) \end{aligned}$$

中文分词的基本算法：CRF概念

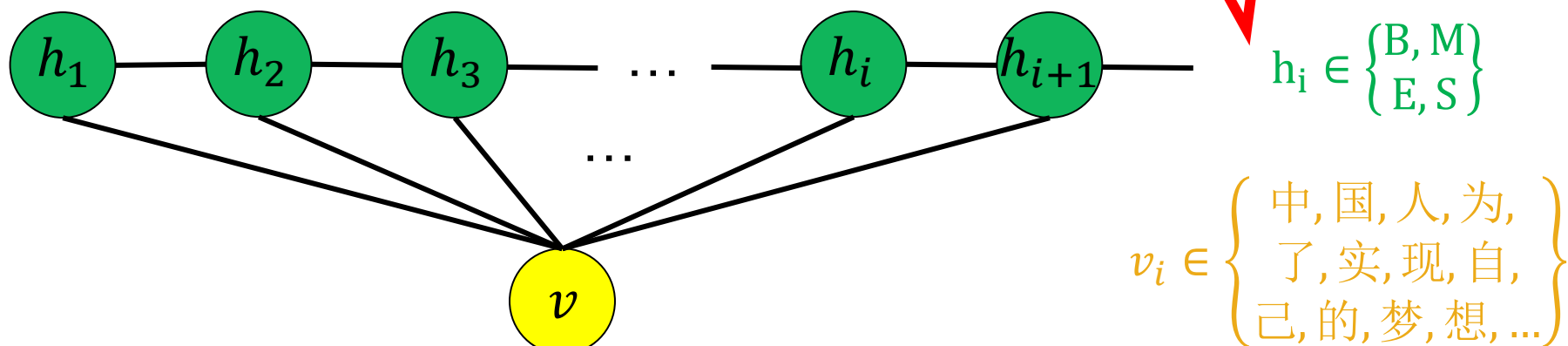


$$\operatorname{argmax}_h P(h|v) = \operatorname{argmax}_h \frac{\operatorname{score}(h, v)}{\sum_h \operatorname{score}(h, v)} \quad (\operatorname{score}(h, v) \geq 0)$$

$$\operatorname{score}(h, v) = \exp \left(\sum_{k=1}^K w_k F_k(h, v) \right)$$

$$\begin{aligned} F_k(h, v) &= F_k(h_1, h_2, \dots, h_i, \dots, h_N, v_1, v_2, \dots, v_i, \dots, v_N) \\ &= \sum_{i=2}^N f_k(h_i, h_{i-1}, v, i) \end{aligned}$$

中文分词的基本算法：CRF概念



$$P(h|v) = \frac{\text{score}(h, v)}{\sum_h \text{score}(h, v)} \quad (\text{s.t.} \quad \text{score}(h, v) \geq 0)$$

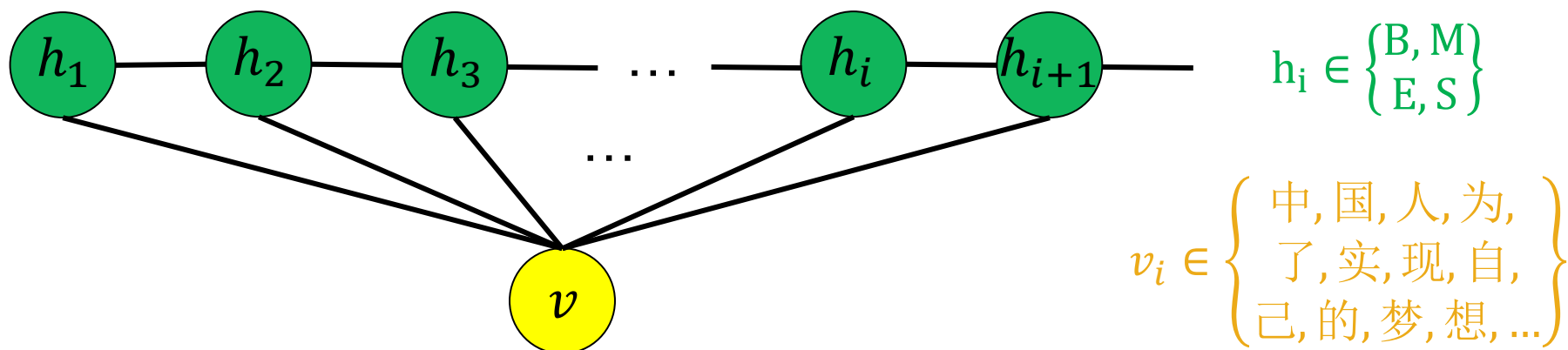
$$\text{score}(h, v) = \exp \left(\sum_{k=1}^K w_k F_k(h, v) \right)$$

$$F_k(h, v) = F_k(h_1, h_2, \dots, h_i, \dots, h_N, v_1, v_2, \dots, v_i, \dots, v_N)$$

$$= \sum_{i=2}^N \left[f_k(h_i, h_{i-1}, v, i) \right]$$

与HMM不同，CRF中允许用户指定具体feature。这使得CRF可以引入各种各样的基于规则的feature

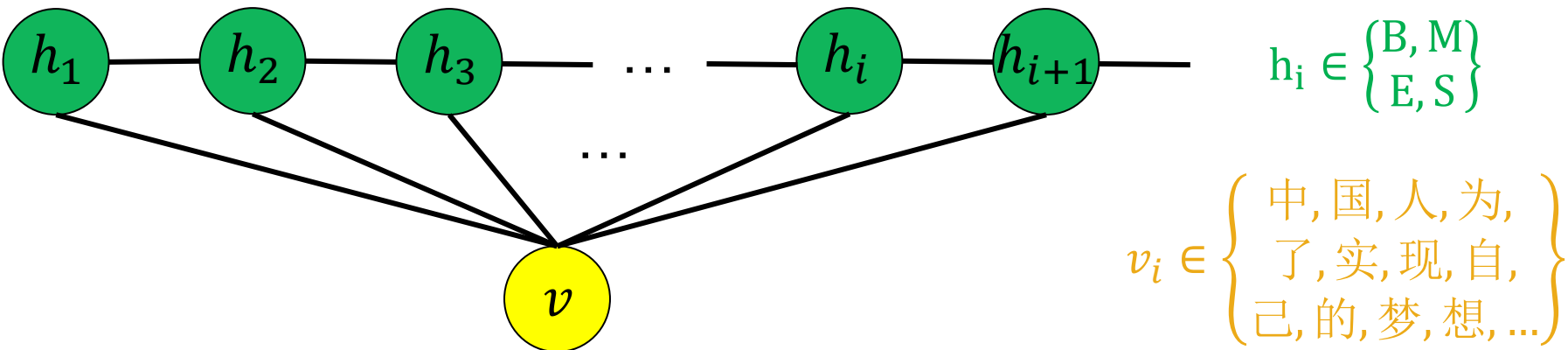
中文分词的基本算法：CRF概念



$f_k(h_i, h_{i-1}, v, i)$ 的选取:

- $f_k(h_i, h_{i-1}, v, i)$ 通常是值域为 $\{0, 1\}$ 的二值函数，在满足特征规定的条件后值为一，否则为零。
- $f_k(h_i, h_{i-1}, v, i)$ 的具体定义可以非常灵活。可以只与输入序列 v 有关，也可以与 h_i, h_{i-1} 和 v 的对应关系有关，甚至可以与feature所在的具体的位置 i 有关。

中文分词的基本算法：CRF概念



$f_k(h_i, h_{i-1}, v, i)$ 的选取:

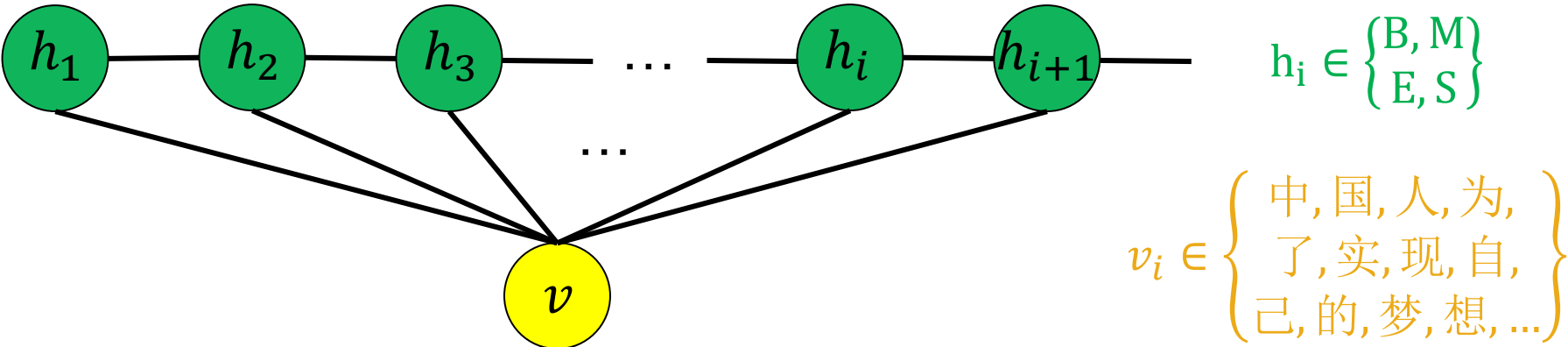
类型	特征	特征激活条件
一元特征	v_{i-1}	上一字符为某特定字符
	v_i	当前字符为某特定字符
	v_{i+1}	下一字符为某特定字符
二元特征	$\overline{v_{i-1}v_i}$	当前字符和上一字符为某特定串
	$\overline{v_i v_{i+1}}$	当前字符和下一字符为某特征串
跳跃特征	$\overline{v_{i-1} \blacksquare v_{i+1}}$	上一字符和下一字符为某特定串
三元特征	$\overline{v_{i-1} v_i v_{i+1}}$	上一字符、当前字符、下一字符为某特定串

.....

.....

.....

中文分词的基本算法：CRF概念



$f_k(h_i, h_{i-1}, v, i)$ 的选取:

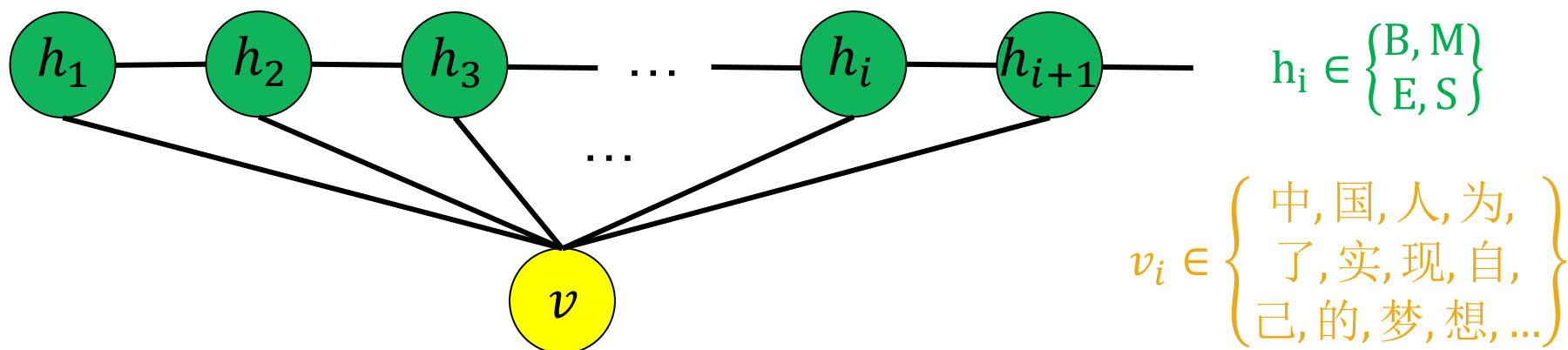
类型	特征	特征激活条件
状态转移	$\overline{h_{i-1}h_i}$	上一字符的标记 h_{i-1} 转移到当前字符的标记 h_i
一元状态特征	h_{i-1}	上一字符为某特定标记
	h_i	当前字符为某特定标记
混合特征	$\overline{v_{i-1}v_i} \cap h_i \cap (i = 5)$	上一字符和当前字符为某特定串, 且 当前字符的标记为 h_i , 且 当前所在位置为5

.....

.....

.....

中文分词的基本算法：CRF概念



$f_k(h_i, h_{i-1}, v, i)$ 的选取:

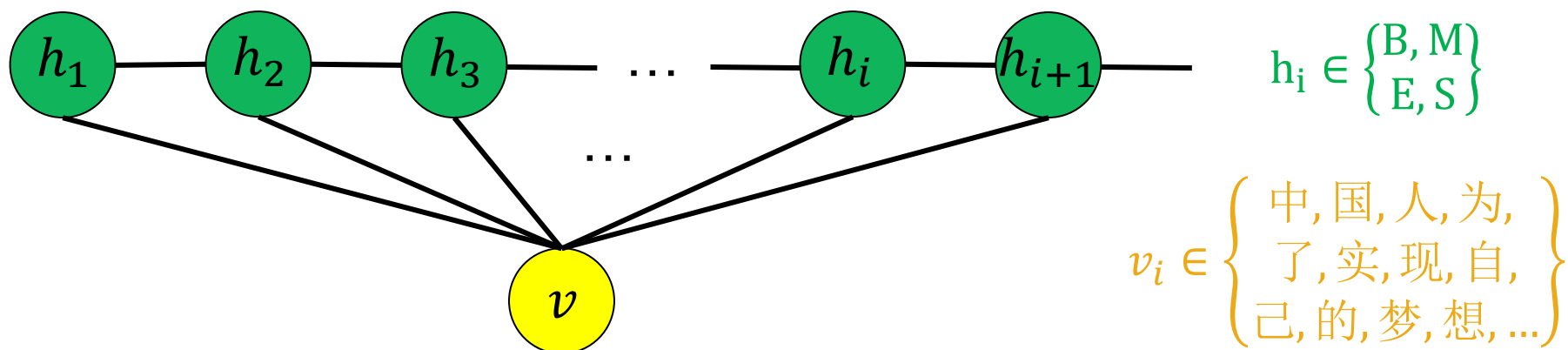
- 对一个具有实用价值的CRF分词器而言, 其使用的**特征 (feature)** 数目 k 在

$10^5 \sim 10^6$

量级。

- 因而, 这些特征多是由**特征模板 (feature template)** 生成的。

中文分词的基本算法：CRF概念



$f_k(h_i, h_{i-1}, v, i)$ 的选取:

- 特征模板 (feature template) :

给定某特征模板, 比如

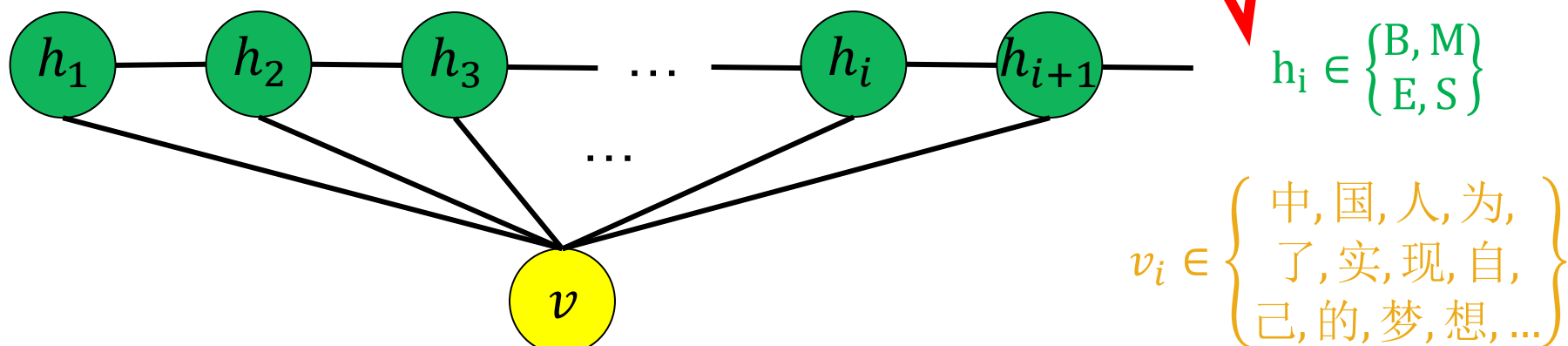
$$\overline{v_{i-1}v_i} \cap h_i \cap (i = 5)$$

算法将依照一定的规则在数据集中遍历, 找到 N 个符合该形式的具体特征。(比如最常出现的 N 个 v_{i-1} , v_i , h_i 和 i 的组合)

- ▶ **什么是中文分词**
 - ▶ 中文分词的规范
 - ▶ 中文分词中的切分歧义
 - ▶ 中文分词中的未登录词
- ▶ **中文分词的几个主要算法**
 - ▶ 最大匹配法
 - ▶ 最短路径法
 - ▶ 语言模型法
 - ▶ 条件随机场 (CRF)
- ▶ **条件随机场 (CRF) 的几个重要算法**
 - ▶ 给定文本推断分词方案: Viterbi算法
 - ▶ 给定文本计算配分函数: 前向与后向算法
 - ▶ 一些有关CRF的推论 (选讲)
 - ▶ 有监督学习: 最大似然参数估计
- ▶ **中文分词的评价指标**

- ▶ **什么是中文分词**
 - ▶ 中文分词的规范
 - ▶ 中文分词中的切分歧义
 - ▶ 中文分词中的未登录词
- ▶ **中文分词的几个主要算法**
 - ▶ 最大匹配法
 - ▶ 最短路径法
 - ▶ 语言模型法
 - ▶ 条件随机场 (CRF)
- ▶ **条件随机场 (CRF) 的几个重要算法**
 - ▶ 给定文本推断分词方案: Viterbi算法
 - ▶ 给定文本计算配分函数: 前向与后向算法
 - ▶ 一些有关CRF的推论 (选讲)
 - ▶ 有监督学习: 最大似然参数估计
- ▶ **中文分词的评价指标**

中文分词的基本算法：CRF概念

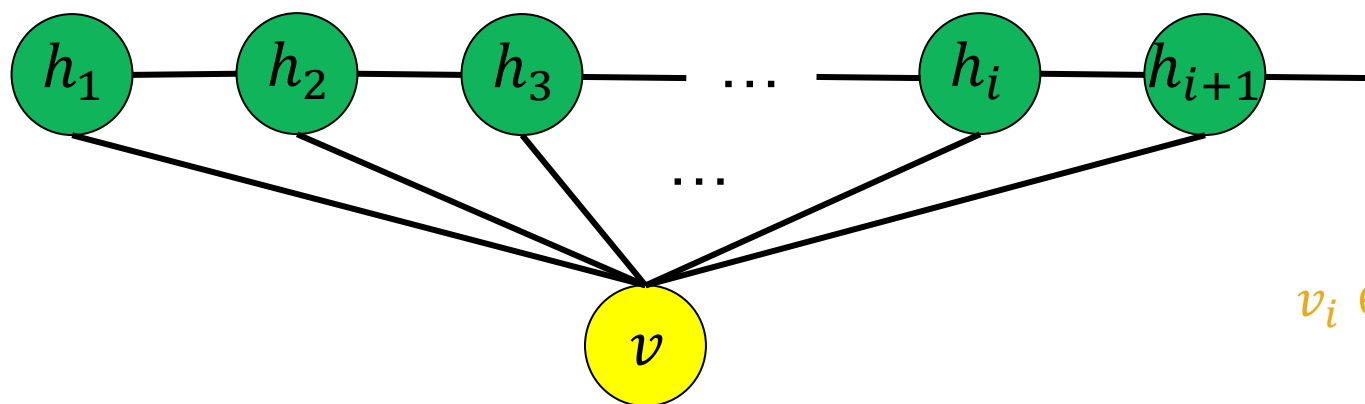


$$P(h|v) = \frac{\text{score}(h, v)}{\sum_h \text{score}(h, v)} \quad (\text{s.t.} \quad \text{score}(h, v) \geq 0)$$

$$\text{score}(h, v) = \exp \left(\sum_{k=1}^K w_k F_k(h, v) \right)$$

$$\begin{aligned} F_k(h, v) &= F_k(h_1, h_2, \dots, h_i, \dots, h_N, v_1, v_2, \dots, v_i, \dots, v_N) \\ &= \sum_{i=2}^N f_k(h_i, h_{i-1}, v, i) \end{aligned}$$

条件随机场：配分函数



$h_i \in \begin{Bmatrix} \text{B, M} \\ \text{E, S} \end{Bmatrix}$

$v_i \in \left\{ \begin{array}{l} \text{中, 国, 人, 为,} \\ \text{了, 实, 现, 自,} \\ \text{己, 的, 梦, 想, ...} \end{array} \right\}$

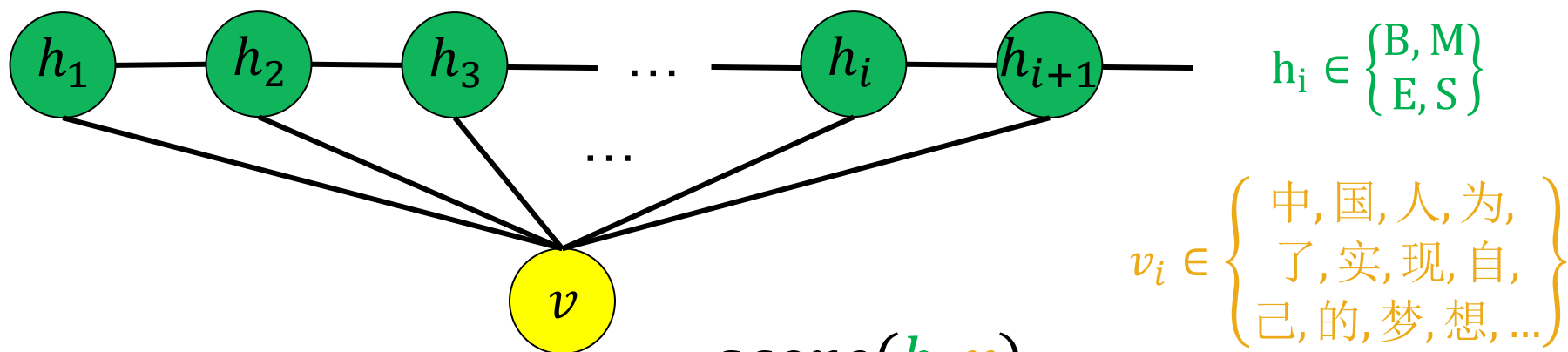
$$P(h|v) = \frac{\text{score}(h, v)}{\sum_h \text{score}(h, v)}$$

($\text{score}(h, v) \geq 0$)

$$\sum_h \text{score}(h, v) = \sum_h \exp \left(\sum_{k=1}^K w_k F_k(h, v) \right) = Z(v)$$

$$Z(v) = \sum_h \exp \left(\sum_{k=1}^K w_k \sum_{i=2}^N f_k(h_i, h_{i-1}, v, i) \right)$$

条件随机场: Viterbi算法

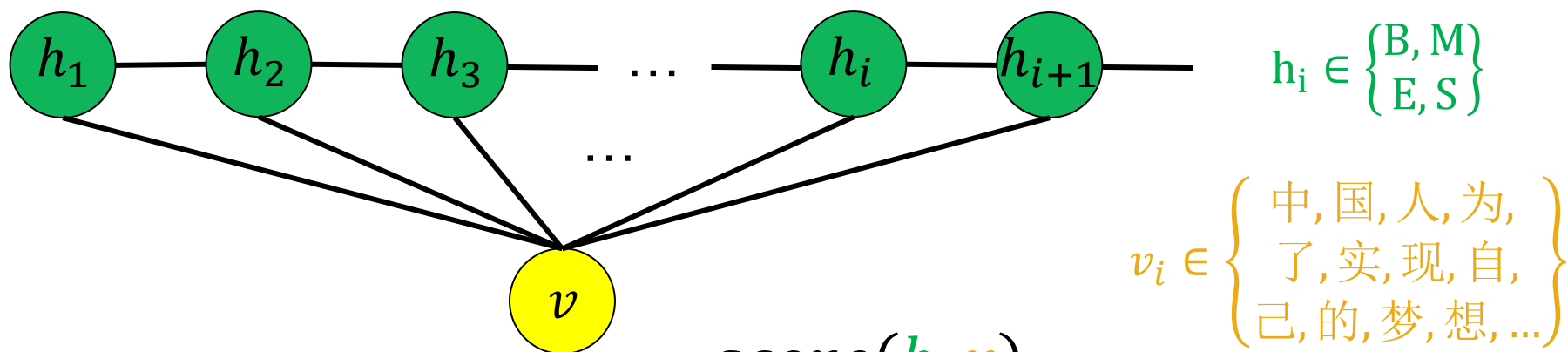


$$\operatorname{argmax}_{\mathbf{h}} P(\mathbf{h}|\mathbf{v}) = \operatorname{argmax}_{\mathbf{h}} \frac{\operatorname{score}(\mathbf{h}, \mathbf{v})}{\sum_{\mathbf{h}} \operatorname{score}(\mathbf{h}, \mathbf{v})} \quad (\operatorname{score}(\mathbf{h}, \mathbf{v}) \geq 0)$$

$$= \operatorname{argmax}_{\mathbf{h}} \frac{\exp\left(\sum_{k=1}^K w_k F_k(\mathbf{h}, \mathbf{v})\right)}{\sum_{\mathbf{h}} \exp\left(\sum_{k=1}^K w_k F_k(\mathbf{h}, \mathbf{v})\right)}$$

$$= \operatorname{argmax}_{\mathbf{h}} \frac{\exp\left(\sum_{k=1}^K w_k F_k(\mathbf{h}, \mathbf{v})\right)}{\cancel{Z(\mathbf{v})}}$$

条件随机场： Viterbi算法

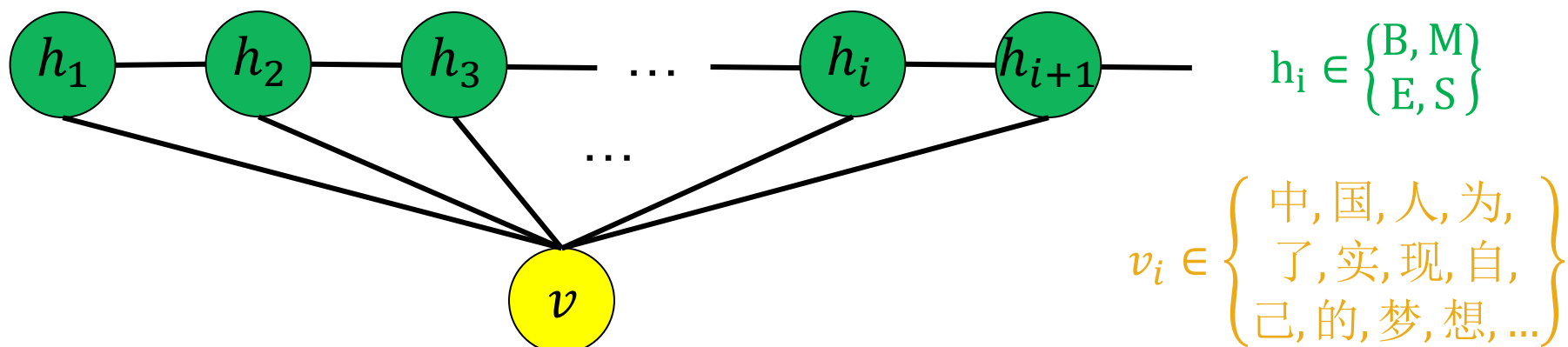


$$\operatorname{argmax}_{\mathbf{h}} P(\mathbf{h}|\mathbf{v}) = \operatorname{argmax}_{\mathbf{h}} \frac{\operatorname{score}(\mathbf{h}, \mathbf{v})}{\sum_{\mathbf{h}} \operatorname{score}(\mathbf{h}, \mathbf{v})} \quad (\operatorname{score}(\mathbf{h}, \mathbf{v}) \geq 0)$$

$$= \operatorname{argmax}_{\mathbf{h}} \frac{\exp\left(\sum_{k=1}^K w_k F_k(\mathbf{h}, \mathbf{v})\right)}{\sum_{\mathbf{h}} \exp\left(\sum_{k=1}^K w_k F_k(\mathbf{h}, \mathbf{v})\right)}$$

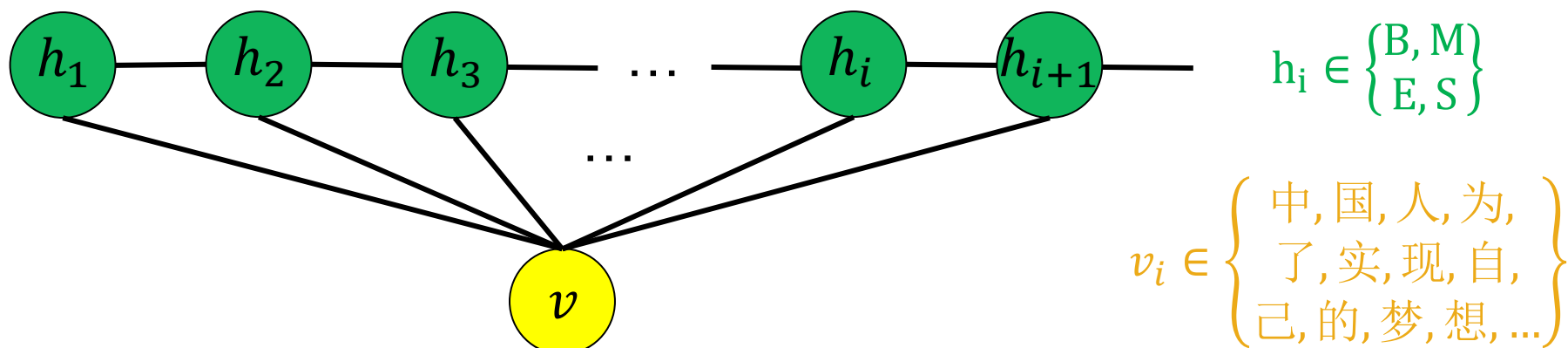
$$= \operatorname{argmax}_{\mathbf{h}} \exp\left(\sum_{k=1}^K w_k F_k(\mathbf{h}, \mathbf{v})\right)$$

条件随机场： Viterbi算法



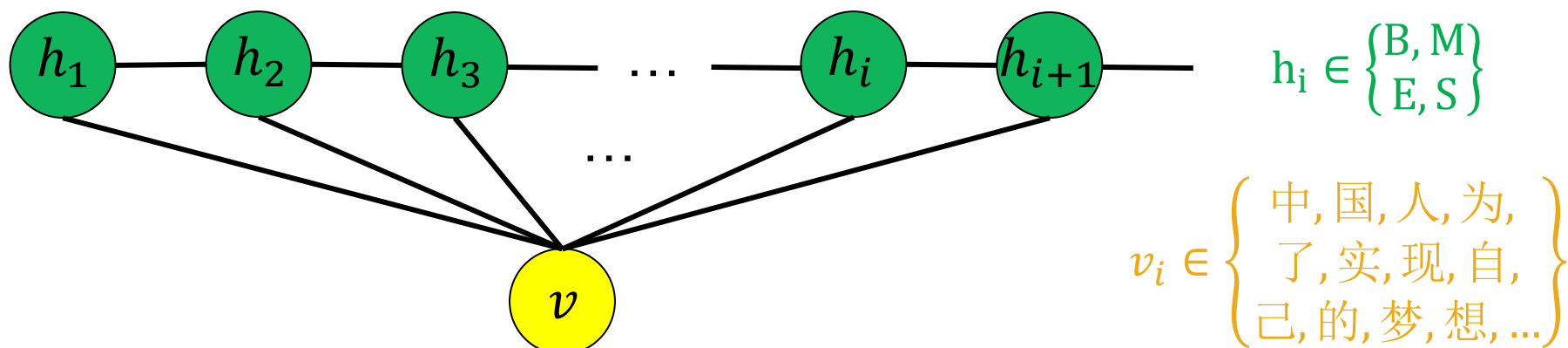
$$\begin{aligned} \operatorname{argmax}_h P(h|v) &= \operatorname{argmax}_h \left[\exp \left(\sum_{k=1}^K w_k F_k(h, v) \right) \right] \\ &= \operatorname{argmax}_h \left[\sum_{k=1}^K w_k F_k(h, v) \right] \\ &= \operatorname{argmax}_h \left[\sum_{k=1}^K w_k \sum_{i=2}^N f_k(h_i, h_{i-1}, v, i) \right] \end{aligned}$$

条件随机场：Viterbi算法



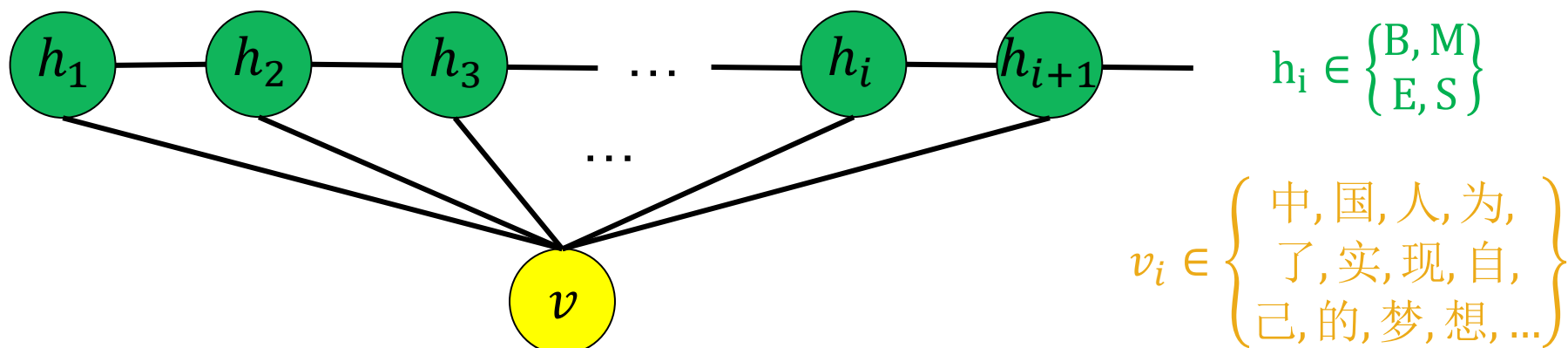
$$\begin{aligned} \operatorname{argmax}_{\mathbf{h}} P(\mathbf{h}|\mathbf{v}) &= \operatorname{argmax}_{\mathbf{h}} \left[\exp \left(\sum_{k=1}^K w_k F_k(\mathbf{h}, \mathbf{v}) \right) \right] \\ &= \operatorname{argmax}_{\mathbf{h}} \left[\sum_{k=1}^K w_k F_k(\mathbf{h}, \mathbf{v}) \right] = \operatorname{argmax}_{\mathbf{h}} \left[\sum_{k=1}^K w_k \sum_{i=2}^N f_k(h_i, h_{i-1}, v, i) \right] \\ &= \operatorname{argmax}_{\mathbf{h}} \left[\sum_{k=1}^K \sum_{i=2}^N w_k f_k(h_i, h_{i-1}, v, i) \right] \end{aligned}$$

条件随机场：Viterbi算法



$$\begin{aligned} \operatorname{argmax}_{\mathbf{h}} P(\mathbf{h}|\mathbf{v}) &= \operatorname{argmax}_{\mathbf{h}} \left[\exp \left(\sum_{k=1}^K w_k F_k(\mathbf{h}, \mathbf{v}) \right) \right] \\ &= \operatorname{argmax}_{\mathbf{h}} \left[\sum_{k=1}^K w_k F_k(\mathbf{h}, \mathbf{v}) \right] = \operatorname{argmax}_{\mathbf{h}} \left[\sum_{k=1}^K w_k \sum_{i=2}^N f_k(h_i, h_{i-1}, v, i) \right] \\ &= \operatorname{argmax}_{\mathbf{h}} \left[\sum_{i=2}^N \sum_{k=1}^K w_k f_k(h_i, h_{i-1}, v, i) \right] \end{aligned}$$

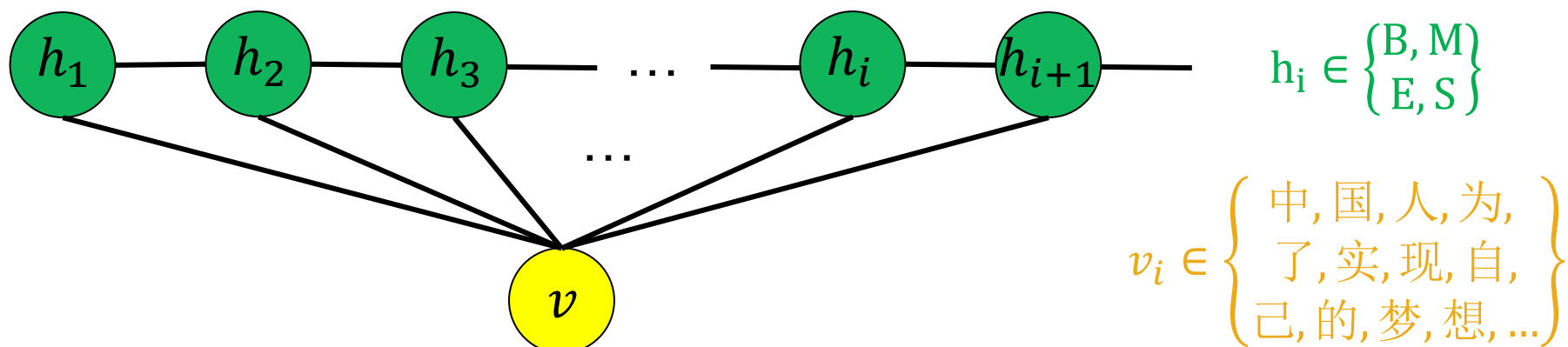
条件随机场: Viterbi算法



$$\operatorname{argmax}_{\underset{h}{h}} P(\underset{h}{h} | \underset{v}{v}) = \operatorname{argmax}_{\underset{h}{h}} \left[\exp \left(\sum_{k=1}^K w_k F_k(\underset{h}{h}, \underset{v}{v}) \right) \right]$$

$$= \operatorname{argmax}_{\underset{h}{h}} \left[\sum_{i=2}^N \sum_{k=1}^K w_k f_k(\underset{h}{h}_i, \underset{h}{h}_{i-1}, \underset{v}{v}, i) \right]$$

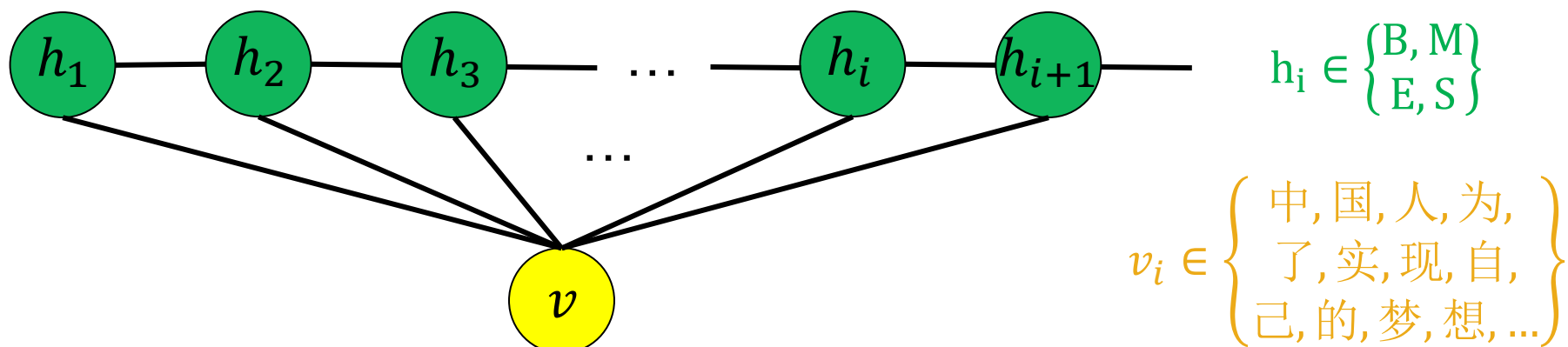
条件随机场：Viterbi算法



$$\operatorname{argmax}_{\mathbf{h}} P(\mathbf{h}|\mathbf{v}) = \operatorname{argmax}_{\mathbf{h}} \left[\exp \left(\sum_{k=1}^K w_k F_k(\mathbf{h}, \mathbf{v}) \right) \right]$$

$$= \operatorname{argmax}_{h_1, h_2, \dots, h_i, \dots, h_N} \left[\sum_{k=1}^K w_k f_k(h_2, h_1, v, 2) + \sum_{k=1}^K w_k f_k(h_3, h_2, v, 3) + \dots \right. \\ \left. + \sum_{k=1}^K w_k f_k(h_N, h_{N-1}, v, N) \right]$$

条件随机场：Viterbi算法



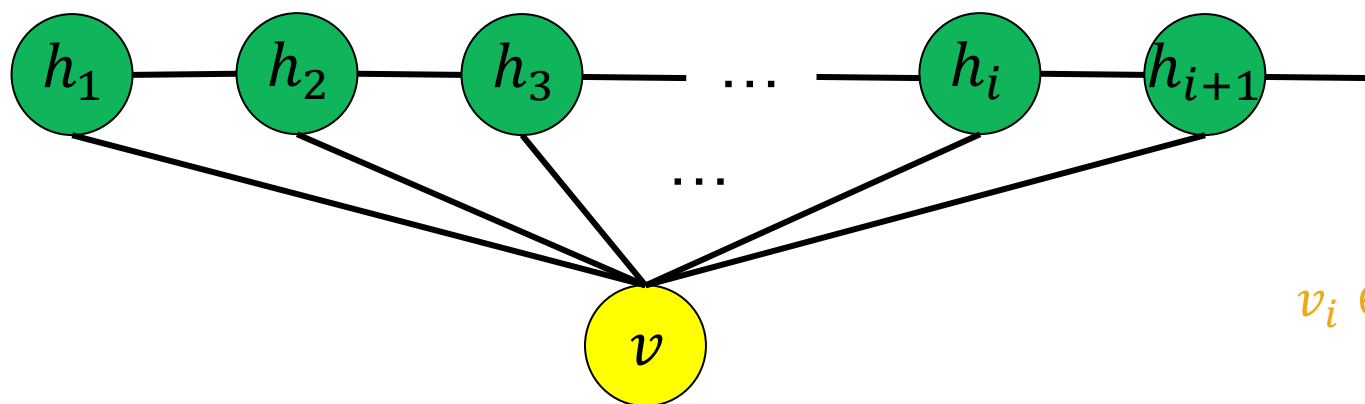
$$\operatorname{argmax}_{\mathbf{h}} P(\mathbf{h}|\mathbf{v}) = \operatorname{argmax}_{\mathbf{h}} \left[\exp \left(\sum_{k=1}^K w_k F_k(\mathbf{h}, \mathbf{v}) \right) \right]$$

$$= \operatorname{argmax}_{h_N} \left\{ \operatorname{argmax}_{h_{N-1}} \left[\sum_{k=1}^K w_k f_k(h_N, h_{N-1}, v, N) + \dots + \operatorname{argmax}_{h_2} \left[\sum_{k=1}^K w_k f_k(h_3, h_2, v, 3) + \operatorname{argmax}_{h_1} \sum_{k=1}^K w_k f_k(h_2, h_1, v, 2) \right] \right] \right\}$$

计算过程与HMM中的Viterbi算法大同小异。

- ▶ **什么是中文分词**
 - ▶ 中文分词的规范
 - ▶ 中文分词中的切分歧义
 - ▶ 中文分词中的未登录词
- ▶ **中文分词的几个主要算法**
 - ▶ 最大匹配法
 - ▶ 最短路径法
 - ▶ 语言模型法
 - ▶ 条件随机场 (CRF)
- ▶ **条件随机场 (CRF) 的几个重要算法**
 - ▶ 给定文本推断分词方案: Viterbi算法
 - ▶ 给定文本计算配分函数: 前向与后向算法
 - ▶ 一些有关CRF的推论 (选讲)
 - ▶ 有监督学习: 最大似然参数估计
- ▶ **中文分词的评价指标**

条件随机场：配分函数



$h_i \in \begin{Bmatrix} \text{B, M} \\ \text{E, S} \end{Bmatrix}$

$v_i \in \left\{ \begin{array}{l} \text{中, 国, 人, 为,} \\ \text{了, 实, 现, 自,} \\ \text{己, 的, 梦, 想, ...} \end{array} \right\}$

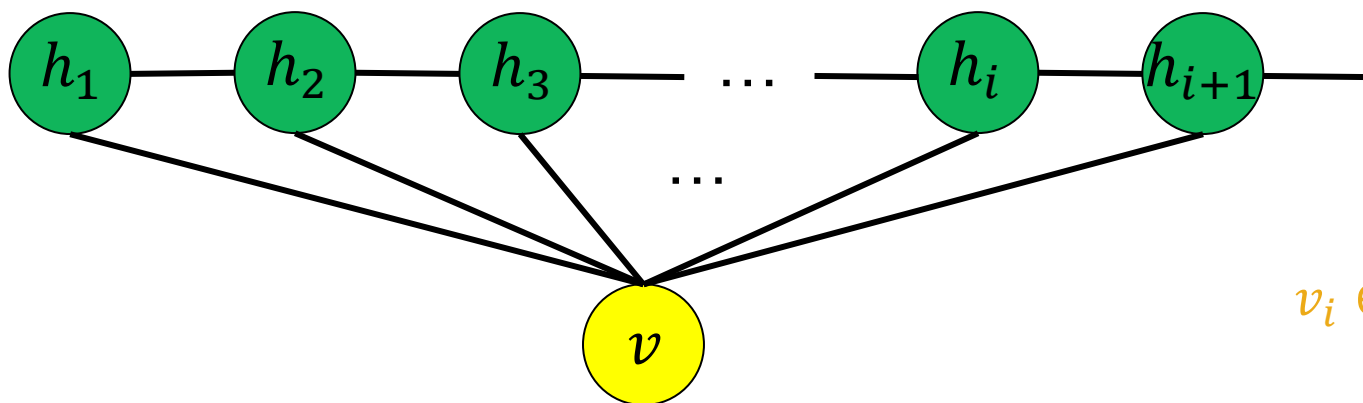
$$P(h|v) = \frac{\text{score}(h, v)}{\sum_h \text{score}(h, v)}$$

($\text{score}(h, v) \geq 0$)

$$\sum_h \text{score}(h, v) = \sum_h \exp \left(\sum_{k=1}^K w_k F_k(h, v) \right) = Z(v)$$

$$Z(v) = \sum_h \exp \left(\sum_{k=1}^K w_k \sum_{i=2}^N f_k(h_i, h_{i-1}, v, i) \right)$$

条件随机场：前向算法与后向算法



$h_i \in \begin{Bmatrix} B, M \\ E, S \end{Bmatrix}$

$v_i \in \begin{Bmatrix} \text{中, 国, 人, 为,} \\ \text{了, 实, 现, 自,} \\ \text{己, 的, 梦, 想, ...} \end{Bmatrix}$

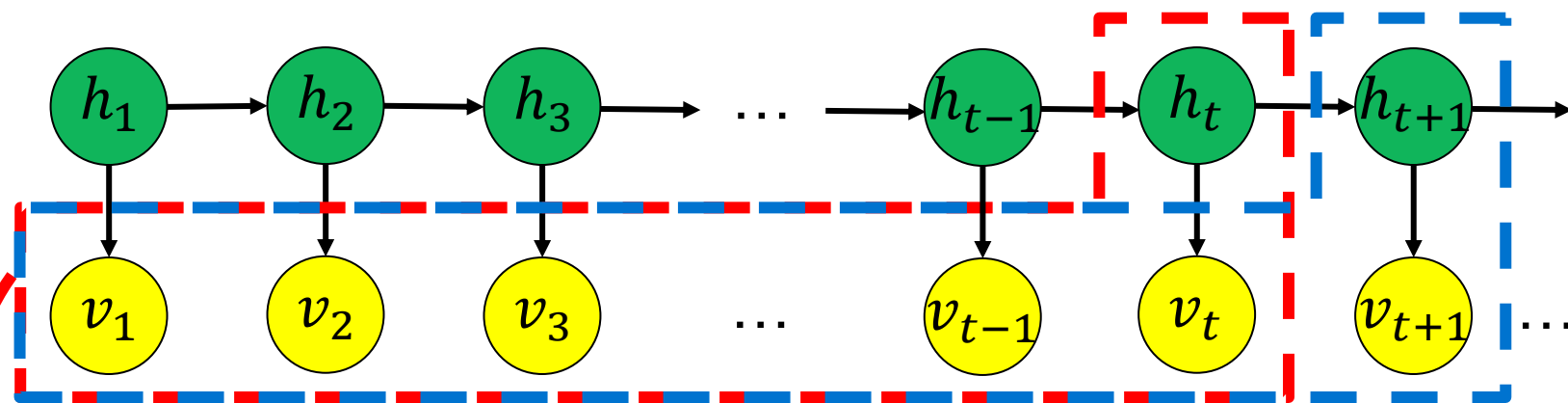
$$Z(v) = \sum_h \exp \left(\sum_{k=1}^K w_k \sum_{i=2}^N f_k(h_i, h_{i-1}, v, i) \right) = \sum_{h_1, h_2, \dots, h_N} \exp \left(\sum_{i=2}^N \sum_{k=1}^K w_k f_k(h_i, h_{i-1}, v, i) \right)$$

$$= \sum_{h_N} \sum_{h_{N-1}} \left\{ \exp \left(\sum_{k=1}^K w_k f_k(h_N, h_{N-1}, v, i) \right) \cdots \sum_{h_2} \left[\exp \left(\sum_{k=1}^K w_k f_k(h_3, h_2, v, i) \right) \sum_{h_1} \exp \left(\sum_{k=1}^K w_k f_k(h_2, h_1, v, 1) \right) \right] \right\}$$

$$= \sum_{h_N} \sum_{h_{N-1}} \left\{ \exp \left(\sum_{k=1}^K w_k f_k(h_N, h_{N-1}, v, i) \right) \cdots \sum_{h_2} \left[\exp \left(\sum_{k=1}^K w_k f_k(h_3, h_2, v, i) \right) \sum_{h_1} \exp \left(\sum_{k=1}^K w_k f_k(h_2, h_1, v, 1) \right) \right] \right\}$$

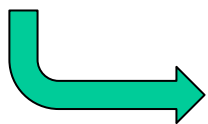
- ▶ **什么是中文分词**
 - ▶ 中文分词的规范
 - ▶ 中文分词中的切分歧义
 - ▶ 中文分词中的未登录词
- ▶ **中文分词的几个主要算法**
 - ▶ 最大匹配法
 - ▶ 最短路径法
 - ▶ 语言模型法
 - ▶ 条件随机场 (CRF)
- ▶ **条件随机场 (CRF) 的几个重要算法**
 - ▶ 给定文本推断分词方案: Viterbi算法
 - ▶ 给定文本计算配分函数: 前向与后向算法
 - ▶ 一些有关CRF的推论 (选讲)
 - ▶ 有监督学习: 最大似然参数估计
- ▶ **中文分词的评价指标**

隐马尔科夫模型： α 与 β



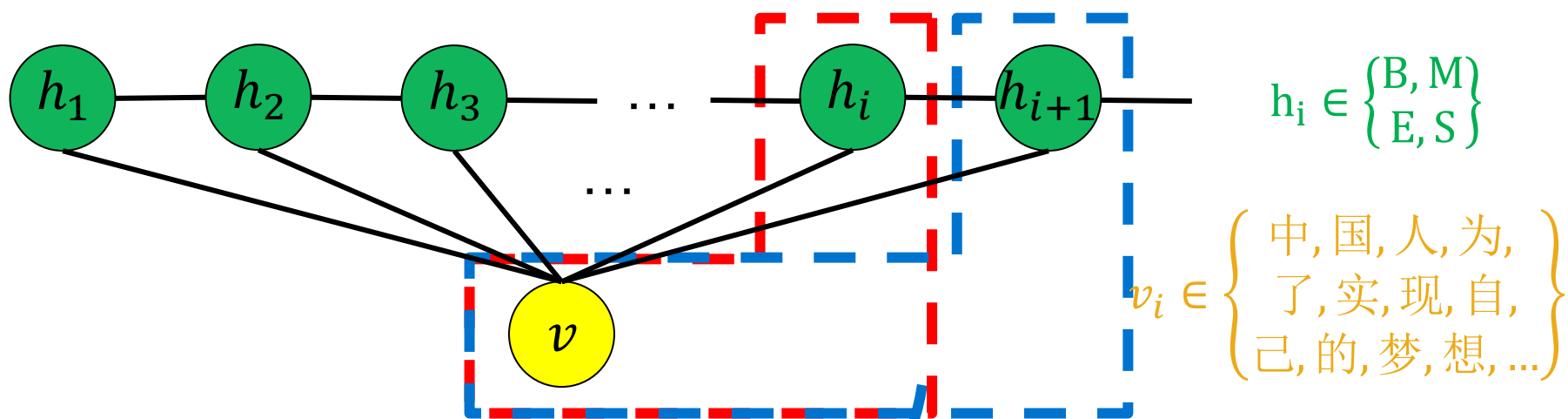
$$P(h, v) = P(v|h) \cdot P(h) = \prod_{j=1}^N P(v_j|h_j) P(h_1) \prod_{i=1}^{N-1} P(h_{i+1}|h_i)$$

$$\alpha_t(j) = P(v_1, v_2, \dots, v_t, h_t = j)$$

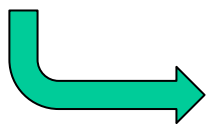


$$\alpha_{t+1}(j) = \sum_{i=1}^S \alpha_t(i) P(h_{t+1} = j | h_t = i) P(v_{t+1} | h_{t+1} = j)$$

隐马尔科夫模型: α 与 β

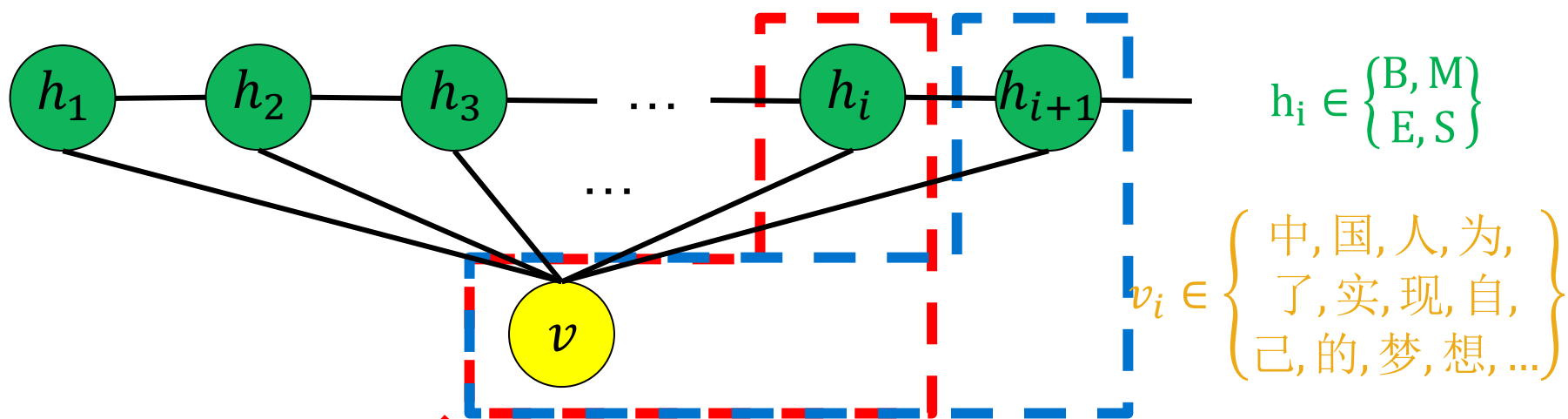


$$\alpha_t(j) = P(v_1, v_2, \dots, v_t, h_t = j)$$



$$\alpha_{t+1}(j) = \sum_{i=1}^S \alpha_t(i) P(h_{t+1} = j | h_t = i) P(v_{t+1} | h_{t+1} = j)$$

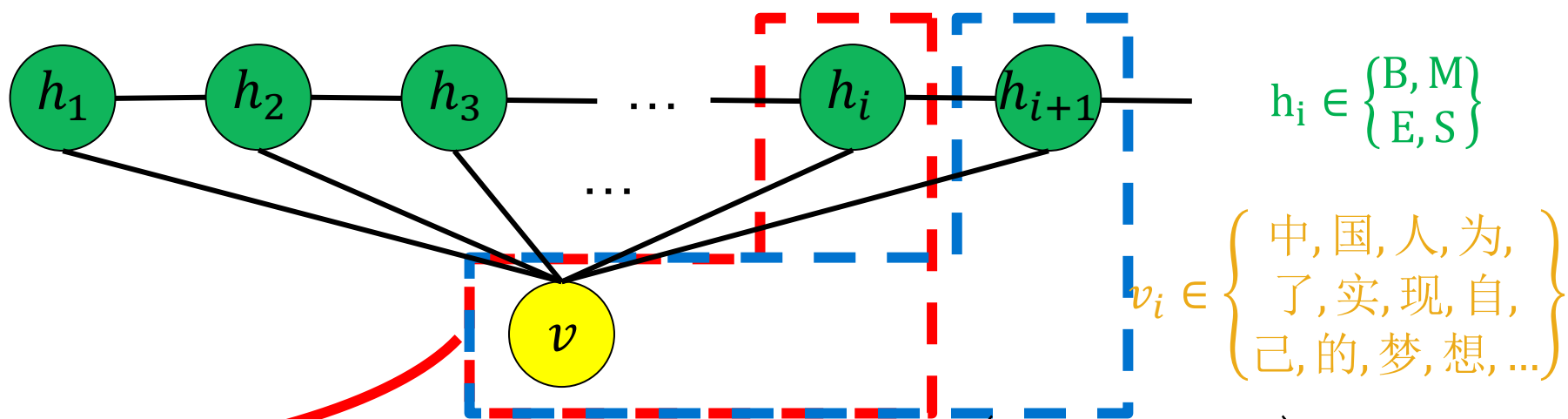
条件随机场: α



$$\alpha_t = P(h_1, h_2, \dots, h_t | v) = \sum_{h_{t+1}, \dots, h_N} \frac{1}{Z(v)} \exp \left(\sum_{k=1}^K w_k F_k(h, v) \right)$$

$$\alpha_{t+1} = P(h_1, h_2, \dots, h_t, h_{t+1} | v) = \sum_{h_{t+2}, \dots, h_N} \frac{1}{Z(v)} \exp \left(\sum_{k=1}^K w_k F_k(h, v) \right)$$

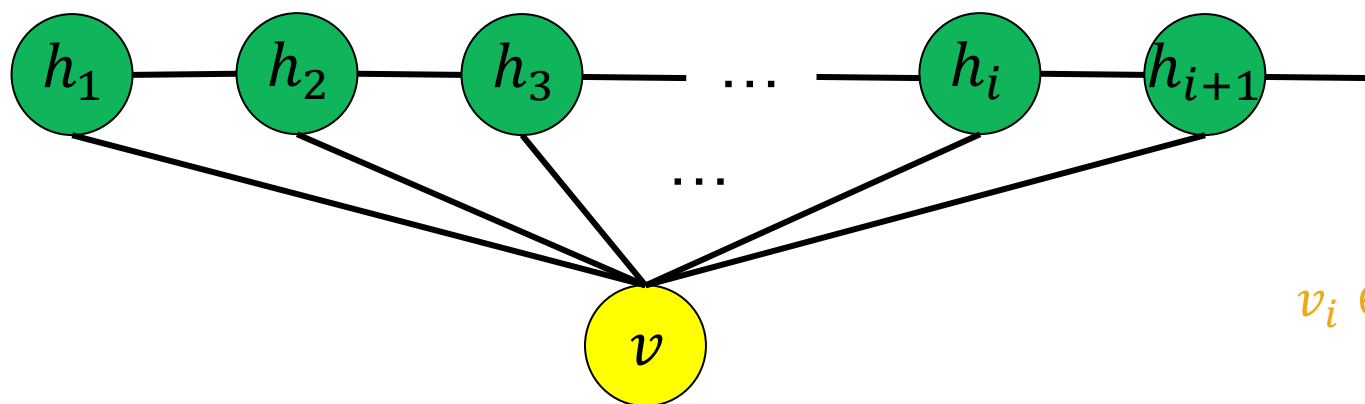
条件随机场: α



$$\begin{aligned} \alpha_t &= P(h_1, h_2, \dots, h_t | v) \\ &= \sum_{h_{t+1}, \dots, h_N} \frac{1}{Z(v)} \exp \left(\sum_{k=1}^K w_k F_k(h, v) \right) \\ &= \sum_{h_{t+1}} \left[\sum_{h_{t+2}, \dots, h_N} \frac{1}{Z(v)} \exp \left(\sum_{k=1}^K w_k F_k(h, v) \right) \right] \\ &= \sum_{h_{t+1}} \alpha_{t+1} \\ \alpha_{t+1} &= P(h_1, h_2, \dots, h_t, h_{t+1} | v) = \sum_{h_{t+2}, \dots, h_N} \frac{1}{Z(v)} \exp \left(\sum_{k=1}^K w_k F_k(h, v) \right) \end{aligned}$$

- ▶ **什么是中文分词**
 - ▶ 中文分词的规范
 - ▶ 中文分词中的切分歧义
 - ▶ 中文分词中的未登录词
- ▶ **中文分词的几个主要算法**
 - ▶ 最大匹配法
 - ▶ 最短路径法
 - ▶ 语言模型法
 - ▶ 条件随机场 (CRF)
- ▶ **条件随机场 (CRF) 的几个重要算法**
 - ▶ 给定文本推断分词方案: Viterbi算法
 - ▶ 给定文本计算配分函数: 前向与后向算法
 - ▶ 一些有关CRF的推论 (选讲)
 - ▶ **有监督学习: 最大似然参数估计**
- ▶ **中文分词的评价指标**

条件随机场:



$$h_i \in \begin{Bmatrix} B, M \\ E, S \end{Bmatrix}$$

$$v_i \in \begin{Bmatrix} \text{中, 国, 人, 为,} \\ \text{了, 实, 现, 自,} \\ \text{己, 的, 梦, 想, ...} \end{Bmatrix}$$

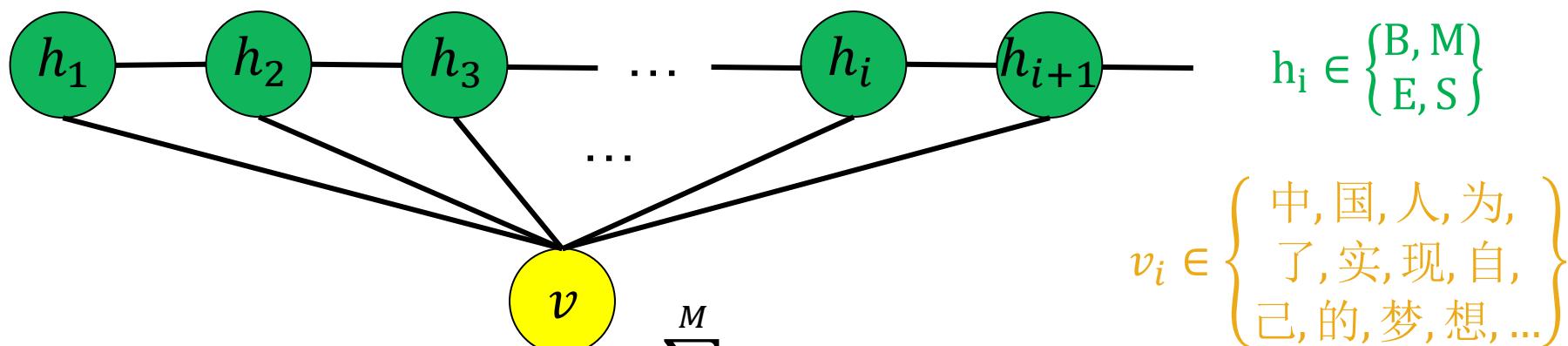
$$P(h|v) = \frac{\text{score}(h, v)}{\sum_h \text{score}(h, v)}$$

$$(\text{score}(h, v) \geq 0)$$

$$\sum_h \text{score}(h, v) = \sum_h \exp \left(\sum_{k=1}^K w_k F_k(h, v) \right) = Z(v)$$

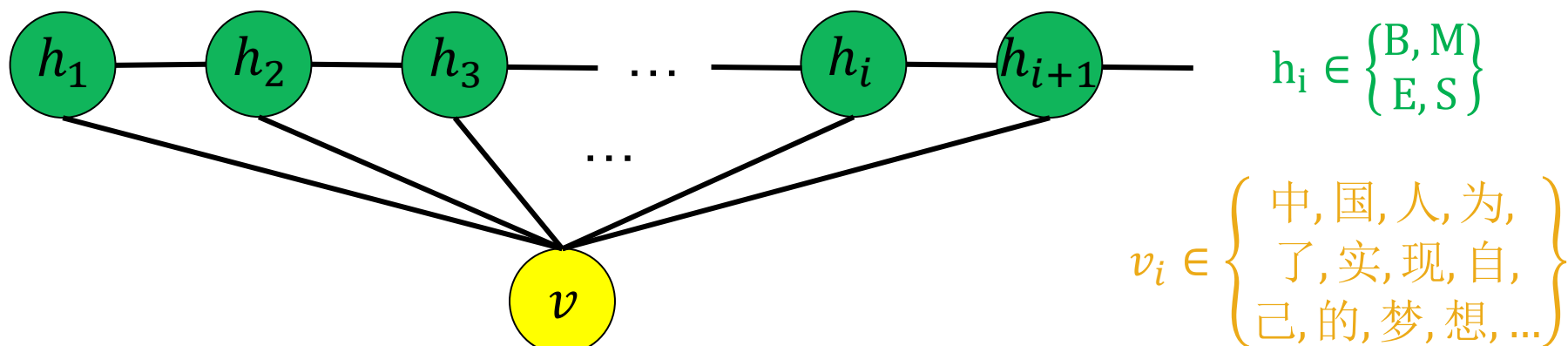
$$Z(v) = \sum_h \exp \left(\sum_{k=1}^K w_k \sum_{i=2}^N f_k(h_i, h_{i-1}, v, i) \right)$$

条件随机场：训练



$$\begin{aligned} \max_w L(w) &= \max_w \sum_{s=1}^M P(h^s | v^s) \\ &= \max_w \sum_{s=1}^M \frac{1}{Z(v^s)} \exp \left(\sum_{k=1}^K w_k F_k(h^s, v^s) \right) \\ &= \max_w \sum_{s=1}^M \frac{1}{Z(v^s)} \exp \left(\sum_{k=1}^K \sum_{i=2}^N w_k f_k(h_i^s, h_{i-1}^s, v^s, i) \right) \end{aligned}$$

条件随机场：训练

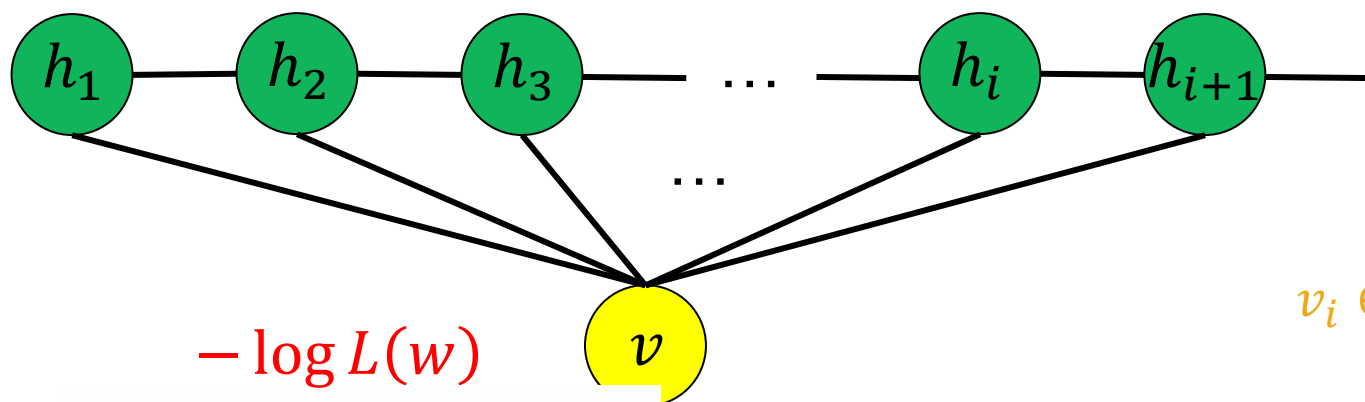


$$\max_w L(w) = \max_w \sum_{s=1}^M \frac{1}{Z(v^s)} \exp \left(\sum_{k=1}^K \sum_{i=2}^N w_k f_k(h_i^s, h_{i-1}^s, v^s, i) \right)$$

$$\Leftrightarrow \max_w [\log L(w)] = \max_w \left[\log \sum_{s=1}^M \frac{1}{Z(v^s)} \exp \left(\sum_{k=1}^K \sum_{i=2}^N w_k f_k(h_i^s, h_{i-1}^s, v^s, i) \right) \right]$$

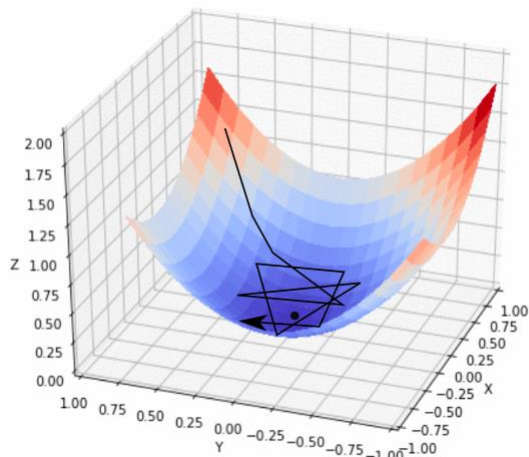
$$\Leftrightarrow \min_w [-\log L(w)] = \min_w \left[-\log \sum_{s=1}^M \frac{1}{Z(v^s)} \exp \left(\sum_{k=1}^K \sum_{i=2}^N w_k f_k(h_i^s, h_{i-1}^s, v^s, i) \right) \right]$$

条件随机场：训练



$$h_i \in \begin{Bmatrix} \text{B, M} \\ \text{E, S} \end{Bmatrix}$$

$$v_i \in \begin{Bmatrix} \text{中, 国, 人, 为,} \\ \text{了, 实, 现, 自,} \\ \text{己, 的, 梦, 想, ...} \end{Bmatrix}$$



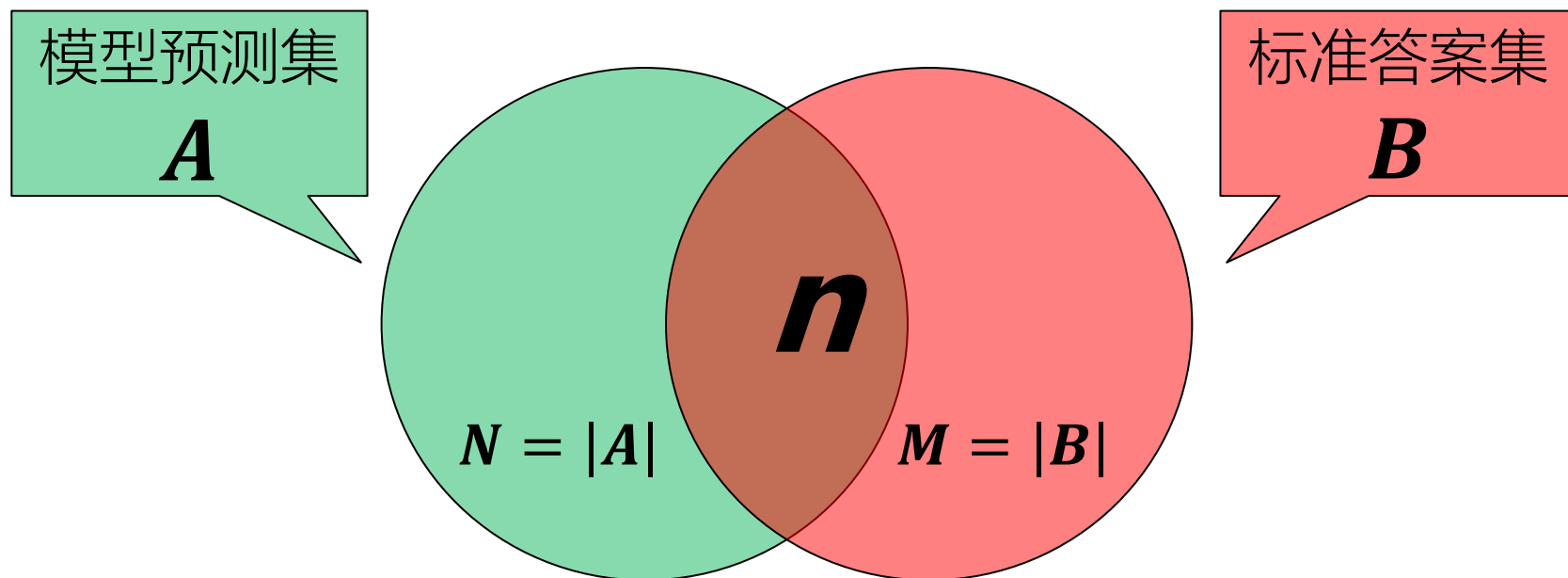
$$w^{p+1} = w^p - \eta \frac{\partial [-\log L(w)]}{\partial w}$$

$$\Leftrightarrow \min_w [-\log L(w)] = \min_w \left[-\log \sum_{s=1}^M \frac{1}{Z(v^s)} \exp \left(\sum_{k=1}^K \sum_{i=2}^N w_k f_k(h_i^s, h_{i-1}^s, v^s, i) \right) \right]$$

- ▶ **什么是中文分词**
 - ▶ 中文分词的规范
 - ▶ 中文分词中的切分歧义
 - ▶ 中文分词中的未登录词
- ▶ **中文分词的几个主要算法**
 - ▶ 最大匹配法
 - ▶ 最短路径法
 - ▶ 语言模型法
 - ▶ 条件随机场 (CRF)
- ▶ **条件随机场 (CRF) 的几个重要算法**
 - ▶ 给定文本推断分词方案: Viterbi算法
 - ▶ 给定文本计算配分函数: 前向与后向算法
 - ▶ 一些有关CRF的推论 (选讲)
 - ▶ 有监督学习: 最大似然参数估计
- ▶ **中文分词的评价指标**

中文分词的评价指标：查准率与查全率

假设系统输出N个结果，其中，正确的结果为n个，标准答案的个数为M个



Precision:

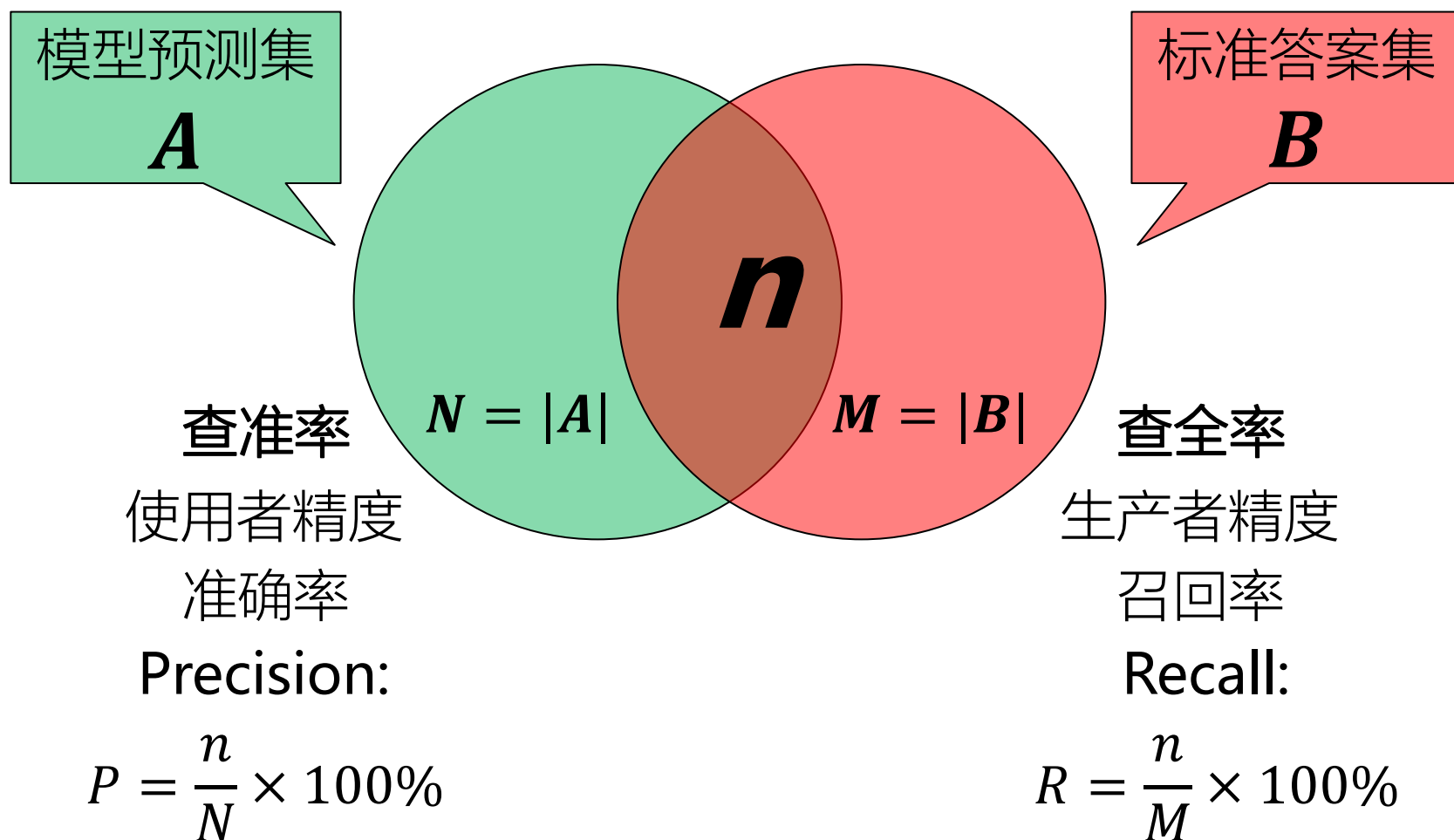
$$P = \frac{n}{N} \times 100\%$$

Recall:

$$R = \frac{n}{M} \times 100\%$$

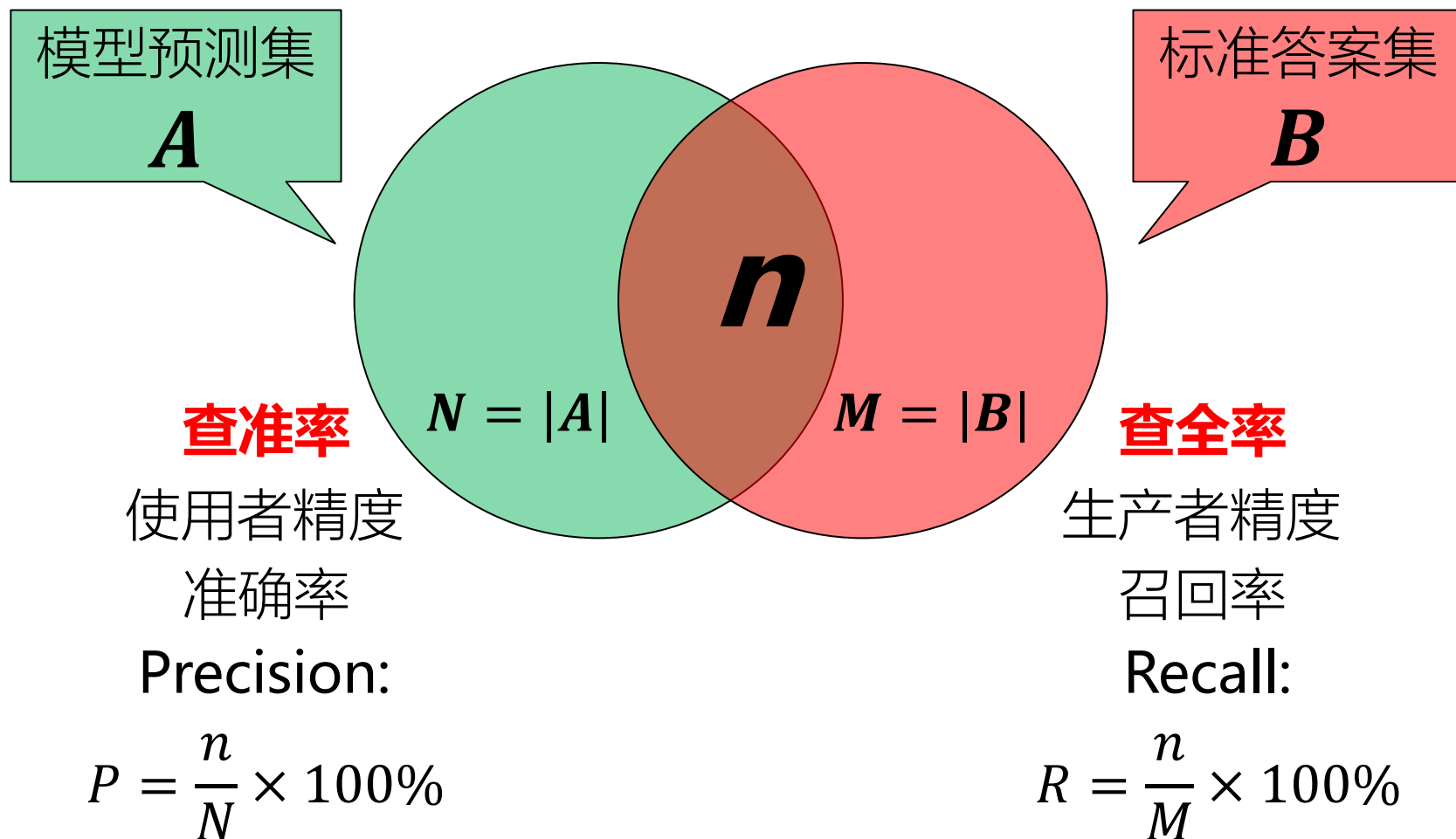
中文分词的评价指标：查准率与查全率

假设系统输出N个结果，其中，正确的结果为n个，标准答案的个数为M个



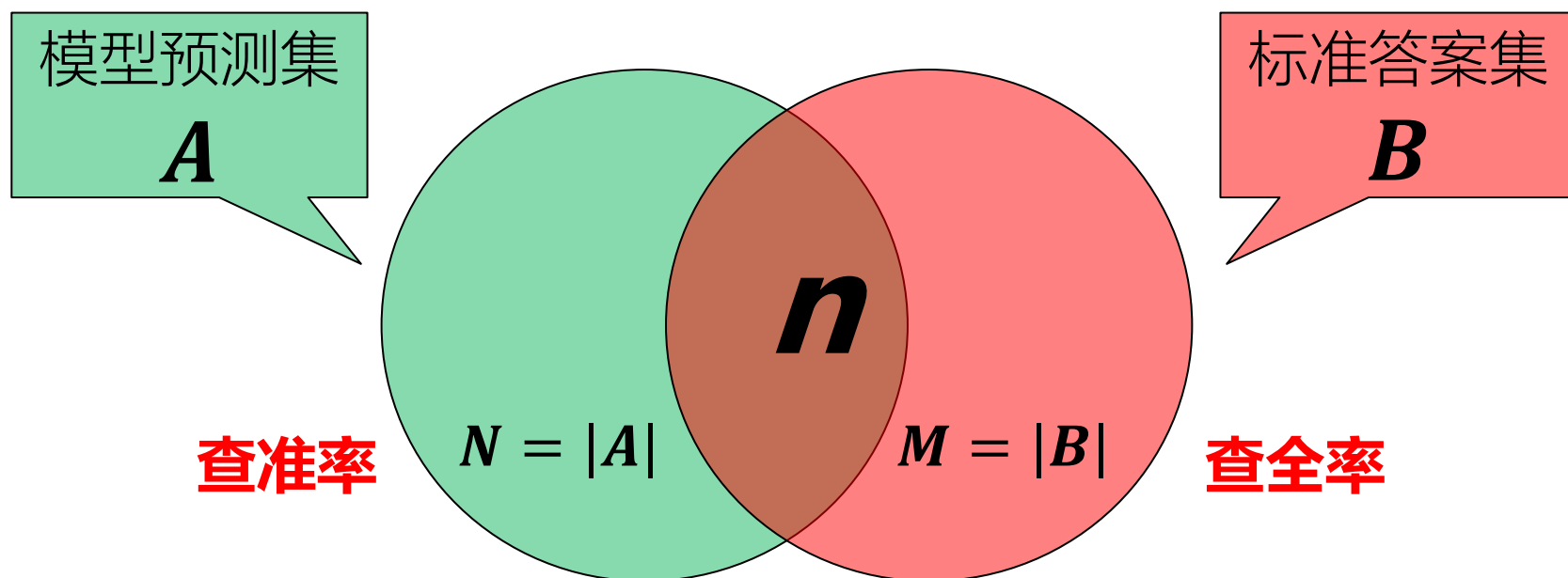
中文分词的评价指标：查准率与查全率

假设系统输出N个结果，其中，正确的结果为n个，标准答案的个数为M个



中文分词的评价指标：F-measure

假设系统输出N个结果，其中，正确的结果为n个，标准答案的个数为M个



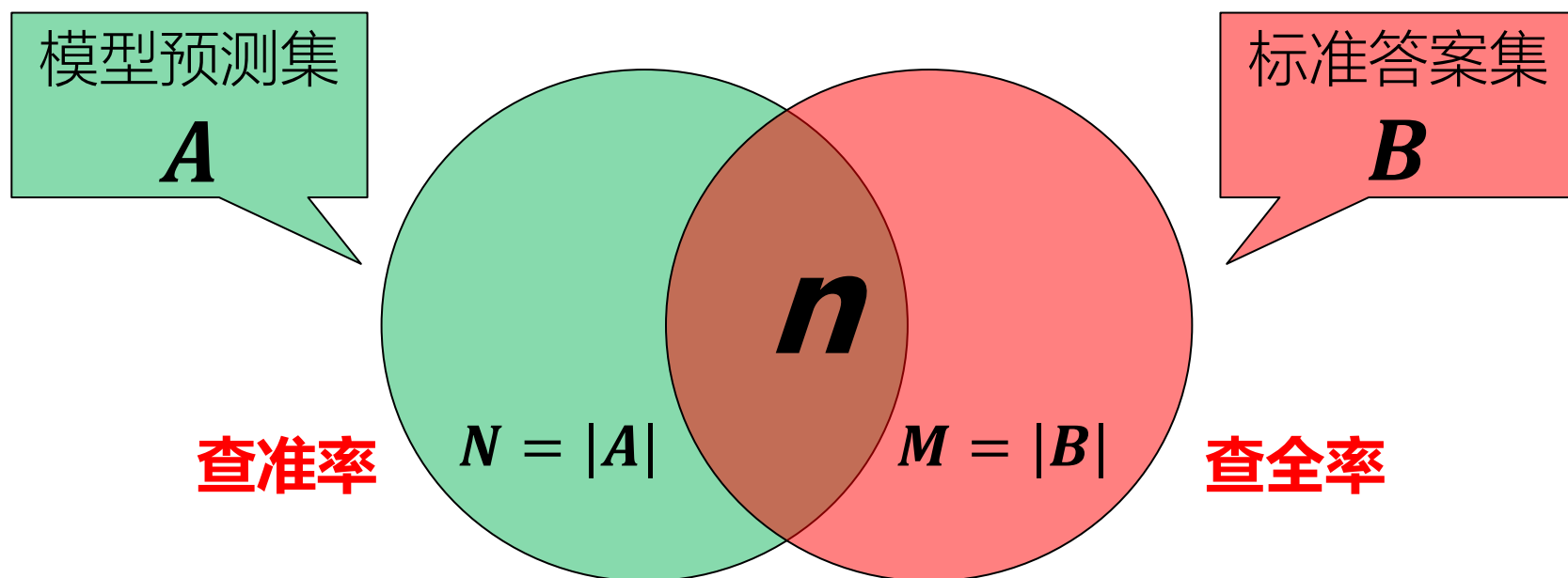
$$F_{\beta} = \frac{(1 + \beta^2)PR}{\beta^2 P + R} \times 100\%$$

$$P = \frac{n}{N} \times 100\%$$

$$R = \frac{n}{M} \times 100\%$$

中文分词的评价指标：F-measure

假设系统输出N个结果，其中，正确的结果为n个，标准答案的个数为M个



$$F_1 = \frac{2PR}{P + R} \times 100\%$$

$$P = \frac{n}{N} \times 100\%$$

$$R = \frac{n}{M} \times 100\%$$

中文分词的评价指标：小测试

假设某个汉语分词系统在一测试集上输出 5260 个分词结果，而标准答案是 4510 个词语，根据这个答案，系统切分出来的结果中有 4120 个是正确的。那么：

$$P = \frac{4120}{5260} \times 100\% = 78.33\%$$

$$R = \frac{4120}{4510} \times 100\% = 91.35\%$$

$$F_1 = \frac{2 \times 78.33\% \times 91.35\%}{78.33\% + 91.35\%} \times 100\% = 84.34\%$$

中文分词：一些实战经验

以SIGHAN Bakeoff 评测语料(2005)为例：

语言模型法 (3-gram)

P=89.8%

CRF

P=94.3%

实用工具：CRF++