南京農業大學
NANJING AGRICULTURAL UNIVERSITY

数字人文下的汉字分词

目录

CONTENT

- 汉语分词基本知识
- 自动分词在数字人文研究中的应用背景
- 语言信息处理中的统计方法

南京農業大學
NANJING AGRICULTURAL UNIVERSITY

汉语自动分词基本知识

什么是自动分词

使用/计算机/将/字符串/自动/转换/为/词串

◆为什么要分词?

●文本分析的第一步

●中文信息处理

●英语、日语

◆分词带来的帮助

●信息检索的预处理：提高查准率

●语音合成的预处理：降低读音复杂性

●汉字识别的后处理：提高识别正确率

●语音识别的后处理：提高识别正确率

●计算机辅助词典编撰：新词、新义项获取

汉语自动分词的三个里程碑

- ◆分词规范（国标、台湾、ISO）
- ◆分词词表（体现各自的分词规范）
- ◆分词竞赛（带标语料库）
  - 搁置“什么是词”的分歧
  - 专注于分词方法，特别是机器学习方法
  - SIGHAN2003以后

分词规范

◆什么是词?

- “词是最小的能够独立活动的有意义的语言成分”—朱德熙
- “词是具有语音形态，又能表示特定意义，且能在句法上单独出现或与其他的词共同形成词组的最小的单位”—汤廷池

◆汉语的词和非词界限不清

- 词还是词素?
  - 楼、院、氧、叶、虎、云、时
- 词还是短语?
  - 鸡蛋、鸭蛋
  - 高射、高射炮、高射机关枪
  - 人造纤维、人造丝、人造革
  - 大型彩色纪录片
  - 多弹头分导重入大气层运载工具

—吕叔湘《汉语语法分析问题

分词规范

分词的前提是确定词语的边界，而语言学关于汉语词语边界的讨论尚无定论。为满足信息处理的需要，全国信息技术标准化技术委员会制定了国标GB/T13715-1992，即《信息处理用现代汉语分词规范》。该规范明确而具体地界定了汉语分词的主题内容和适用范围，并相对全面地规定了分词原则，在一定程度上有效地保证了各种汉语信息处理系统之间的兼容性。其规定汉语分词的对象包含了词和词组，并定义了分词单位的概念以指示上述对象。在国标GB/T13715-1992的基础上，一些研究机构从自然研究的需求出发，也制定了相应的规范，比如面向通用领域的南京师范大学分词规范、面向新时代人民日报语料的南京农业大学自动分词规范、面向中国古代典籍跨语言文本的南京农业大学中国古代典籍跨语言自动分词规范等。

信息处理用现代汉语分词规范

GB/T 13715-92 文档

主题内容和适用范围（规定分词原则，满足信息处理需要，规范汉语信息处理，兼容各种汉语信息处理系统；汉语信息处理各领域可根据需要加以补充和细化）

◆引用标准

◆术语（汉语信息处理、词、词语、分词单位、汉语分词）

◆概述（10项原则）

◆具体说明（分13种词类叙述细则）

- 名词、动词、形容词、代词、数词、量词、
- 副词、介词、连词、助词、语气词、叹词、象声词

分词规范的10项原则

1. 标点符号是分隔标记；

2-4. 词长一般为二至四字；结合紧密、使用稳定的二至四字词组，一律为分词单位；五字以上结合紧密、使用稳定的谚语、格言拆开后违背原意或影响后续处理，也作为分词单位；

5-9. 惯用语、略语、儿化词、非汉字符号、音译外来词都作为分词单位；

10. 同形异构的，根据上下文做不同切分。

分词单位：汉语信息处理使用的、具有确定的语义或语法功能的基本单位（词或凝固短语）

分词规范具体问题讨论

- 绿/叶，小/床
- 我们，你们，他们，人们，朋友/们，学生/们
- 五月，元月，3月，1988/年/3/月/15日
- 汉族，哈萨克/族，长江，牡丹江，乌苏里/江，忻县，正定/县，黄山，横断山，沂蒙/山
- 学院路，刘家村，永久/牌，中华/烟，牡丹/Ⅲ/型
- 说说/看，研究/研究，想/一/想，想/了/想，想/了/一/想
- 看/不/看，相信/不/相信，容易/不/容易
- 七/百/二/十/三，五/分之/三，百/分之/二
- 生/于，走/向/胜利

分词单位

根据《信息处理用现代汉语分词规范》中的定义，分词单位即“汉语信息处理使用的、具有确定的语义或语法功能的基本单位。它包括本规范的规则限定的词和词组。” 此外，《信息处理用现代汉语分词规范》 中的词为“最小的能独立运用的语言单位”，词组为“由两个或两个以上的词，按一定的语法规则组成，表达一定意义的语言单位”，比如“聂/海胜/谈/中国/航天员/的/未来/”这个现代汉语中共有7个词，而“中国”和“航天员”为两个词，而“中国航天员”则为一个词组，同样的在“古者/富/贵/而/名/摩/灭/，/不/可/胜/记/，/唯/倜傥/非/常/之/人/称/焉”这个古汉语例子中共有21个词，其中“胜”和“记”为两个词，而“胜记”则为一个词组。

自动分词的主要方法

- ◆机械切分：正向/逆向最大匹配法（速度快、实用）
- ◆简单的统计方法：最大概率（N元模型）
- ◆当前主流：统计机器学习
  - 隐马尔科夫（HMM）
  - 最大熵（ME）
  - 支持向量机（SVM）
  - 条件随机场（CRF）
  - 深度学习（RNN/LSTM）

最大匹配法

分词思想：长度最小的词串是最佳词串。
匹配：将汉字串跟词表中的词进行比较。
最大：长词优先，或称“最少分词法”。

- 词表：游戏、公司、天堂、任天堂
- 任天堂/游戏/公司

而不切分为：

- 任/天堂/游戏/公司

长词优先原则在绝大多数情况下是对的。

- *研究/生命/科学

最大匹配法的要点

◆词表：词语的静态查找表，是关于“什么是词”的明确定义，不需要词频数据，也不必将单字词列入。

◆最大词长：词表中最长词的长度，以字符为单位计算。

◆候选词：从某位置开始截取的一个字符串，初始长度为MIN(最大词长,剩余串长)。

- 候选词在底表中查找成功，便确定为词，找不到则将候选词末尾减一字，继续查找。
- 候选词长度为1时不必查找，默认为词。

最大匹配法示例（最大词长 = 4）

输入字符串：他没想到绝地求生过气了

输出词串：无（没找到）

最大匹配法示例（最大词长 = 4）

输入字符串：他没想到绝地求生过气了

输出词串：无（没找到）

最大匹配法示例（最大词长 = 4）

输入字符串：他没想到绝地求生过气了

输出词串：无（没找到）

最大匹配法示例（最大词长 = 4）

输入字符串：他没想到绝地求生过气了

输出词串：他/（单字成词）

输入字符串：没想到绝地求生过气了

最大匹配法示例（最大词长 = 4）

输入字符串：没想到绝地求生过气了

输出词串：他/

最大匹配法示例（最大词长 = 4）

输入字符串：没想到绝地求生过气了

输出词串：他/

最大匹配法示例（最大词长 = 4）

输入字符串：没想到绝地求生过气了

输出词串：他/没/

输入字符串：想到绝地求生过气了

最大匹配法示例（最大词长 = 4）

输入字符串：想到绝地求生过气了

输出词串：他/没/

最大匹配法示例（最大词长 = 4）

输入字符串：想到绝地求生过气了

输出词串：他/没/

最大匹配法示例（最大词长 = 4）

输入字符串：想到绝地求生过气了

输出词串：他/没/想到/（在词表中找到）

输入字符串：绝地求生过气了

最大匹配法示例（最大词长 = 4）

输入字符串：绝地求生过气了

输出词串：他/没/想到/绝地求生/

输入字符串：过气了

最大匹配法示例（最大词长 = 4）

输入字符串：过气了

输出词串：他/没/想到/绝地求生/

最大匹配法示例（最大词长 = 4）

输入字符串：过气了

输出词串：他/没/想到/绝地求生/过气/

输入字符串：了

最大匹配法示例（最大词长 = 4）

输入字符串：了

输出词串：他/没/想到/绝地求生/过气/了

输入字符串：空（停止）

正向匹配与逆向匹配

- 正向匹配：从串首开始做最大匹配，直到串尾。
- 逆向匹配：从串尾开始做最大匹配，直到串首。
- 据报道，逆向最大匹配比正向最大匹配的正确率要略高一些。

正向、逆向匹配的配合

◆正向匹配错误，逆向匹配正确例：

使/用户/满意

◆正向匹配正确，逆向匹配错误例：

市场/需求/和/规格/说明

●可以凭借正向匹配和逆向匹配的结合来发现绝大部分分词歧义。

◆正向、逆向匹配皆错例：

从/马/上/跳/下来

最大匹配法的优劣

◆优点

- 速度快、直观
- 切分一致度高

◆缺点

- 依赖词表
  - 几乎无法解决未登录词问题（只能猜对未登录的单字）
  - 跨领域性较差
- 分词精度不高（85%左右）
- 无法解决切分歧义
  - 交集型歧义：只能根据频率猜
  - 组合型歧义：只合不分

中文自动分词的三大难题

- ◆未登录词：自动分词主要是根据底表来进行的，真实文本中存在大量的未见于底表的词语，它对自动分词正确率的影响最大。
- ◆分词歧义：根据底表，一个串可以切开也可以不切开（组合性歧义），或者可以切在这里也可以切在那里（交集型歧义），但从上下文来看，至少有一种切法是不正确的。
- ◆分词不一致：上下文相同或相似情况下，一个串在分词语料库中有多种切法，也许几种切法都有道理，但应该保持一致。

未登录词

汉语自动分词需要依据底表来辅助模型构建或评价分词结果。然而，真实文本中存在大量的未见于底表的词语，对自动分词的准确率造成很大影响。未登录词不可能被穷尽，且语言的变化和发展始终会带来新的未登录词（如网络流行词语），比如“奥利给、内卷、杠精”等。因此未登录词的切分是汉语分词需要解决的重要问题，未登录词的切分效果也是衡量汉语分词性能的一个重要指标。

未登录词

◆未登录词：out-of-vocabulary（OOV）

- 小猪佩奇身上纹，掌声送给社会人。
- 好嗨呦！感觉人生已经达到了巅峰！
- 我们都是佛系青年，偶尔会转发个锦鲤，上网只相信官宣

◆未登录词不可能被穷尽

切分歧义

根据底表，一个待分词汉字串可能会具有多种分词切分形式，从构成分词歧义。分词歧义一般可以归纳为两类。一类是组合型歧义，即待分词汉字串（一般为两个汉字）既可以切开也可以不切开，如“从马上跳下来”中的“马上”；另一类是交集型歧义，即待分词汉字串（至少三个汉字）有多个切分位置，如“使用户满意”中的“使用户”。一般情况下，可以根据上下文消解分词歧义 。

分词的一致性问题

上下文相同或相似情况下，存在同类分词歧义的待分词汉字串应该始终保持切分方式的一致。对于人工标注分词语料或机器自动分词结果来说，分词一致性都是衡量分词质量的重要指标。人工标注分词语料库构建时可以通过多组多轮交叉验证的形式保证一致性；机器自动分词则应在模型构建时充分考虑分词一致性的问题。在所构建的语料库中分词不一致的现象会出现，比如“中国科学院”存在“中国科学院/”和“中国/科学院/”这种分词形式。

分词的一致性问题

◆个例的不一致
- 发展中国家/，发展中/国家/，发展/中/国家/

◆类型的不一致
- 教育部/， 林业/部/
- 露出（合/总：94.4%）揭开（合/总：87.5%）
- 翻开（合/总：34.8%）离去（合/总：28.6%）

◆跟交集型歧义和组合型歧义都有纠缠：
- 他/把/花鸟画/成/了 一/团浓墨
- 希望/尽快/将/这/幅/作品/画成
- 希望/尽快/将/这/幅/作品/画/成

◆训练语料中本身存在不一致

语料库中多种切分形式的类型分布

	型数	比例(%)	例数	比例(%)
切分不一致	1034	58.09	14254	31.23
组合型歧义	422	23.71	19454	42.62
两者混合	165	9.27	10730	23.51
专名	156	8.76	1199	2.63
切分错误	3	0.17	6	0.01
总数	1780	100.00	45643	100.00

- 语料为1998年1月人民日报文本，董宇统计
- 从中可以看出，从型数看，切分不一致占一半以上
- 但从例数来看，组合型歧义占的比例最大，但切分不一致也占了近三分之一，可见问题的严重性

用概率的乘法定理来计算句子概率

- 句子（词串）是一个状态序列：
- S = w1w2…wn - 1wn
- 由概率的乘法定理可得一个句子的概率为：
- P(S)=P(w1w2…wn - 1wn)
- =P(w1)P(w2|w1)P(w3|w1w2) … P(wn|w1w2…wn-1)
- 计算量太大，数据稀疏问题严重，特别是最后一个词的转移概率计算需要考虑前面所有的词。

N元模型（Ngram）

- N元模型认为，状态序列中的某个状态是否出现，只与它前面的N－1个状态有关（马尔科夫假设）。
- N元模型求序列W的概率时，是在概率乘法定理基础上的简化。大大减少了计算量，缓解了数据稀疏问题。
- 马尔科夫假设并不完全符合语言实际。这既是模型的一个缺点，但同时也是为了实用而付出的必要代价。

自动分词的一元模型

- 一个字串可对应于若干个词串。
- 词串的概率等于每个词的概率的乘积。
- 假设：概率最大的词串是所对应字串的最好的切分形式。
- 这是一种概率分词算法：最大概率法

$$W' = \arg\max_{W} P(W) = \arg\max_{W} P(w_1)P(w_2)\ldots P(w_n)$$

$$= \arg\min_{W} \sum_{i=1}^{n} -\log(P(w_i))$$

一元模型分词的数据准备

- 一个较大规模的分词语料库，从训练集里经统计得到一个词频表。一般而言，训练集规模应该在百万词次以上。
- 如果没有分词语料库，或者规模较小，可以使用一个未经分词的原始语料库，统计其中长度为1～K（K为最大词长）的字串的频率，用串频来近似词频。

一元模型分词示例

- ◆结合成分子时
- ◆切分方式
  - 结合/成/分子/时
  - 结合/成分/子时
  - 结/合成/分子/时
  - 结/合成/分/子时
  - ……
- ◆求解方法
  - 词串概率乘积最大的切分方式
  - 取-log变成费用，词串费用之和最小的切分方式

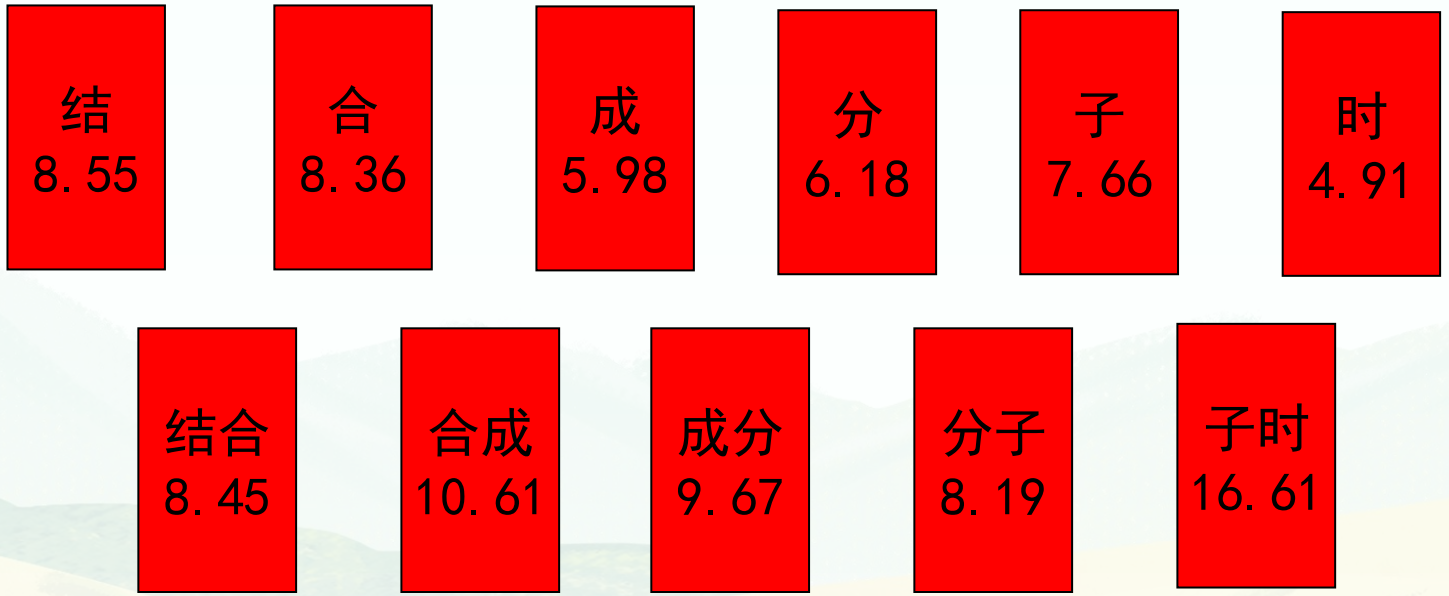
更便捷的方法：动态规划算法

- 使用动态规划算法的目的是减少计算量。
- 获取汉字串中全部候选词及其概率并转为费用；
- 对每个非首词的候选词，找出累计费用最小的前驱词作为最佳前驱词，该候选词的累计费用是它本身的费用加上最佳前驱词的累计费用；
- 对每个结尾的候选词，选择累计费用最小者作为最佳路径上的尾词；
- 从尾词开始，根据最佳前驱线索逆推最佳路径。

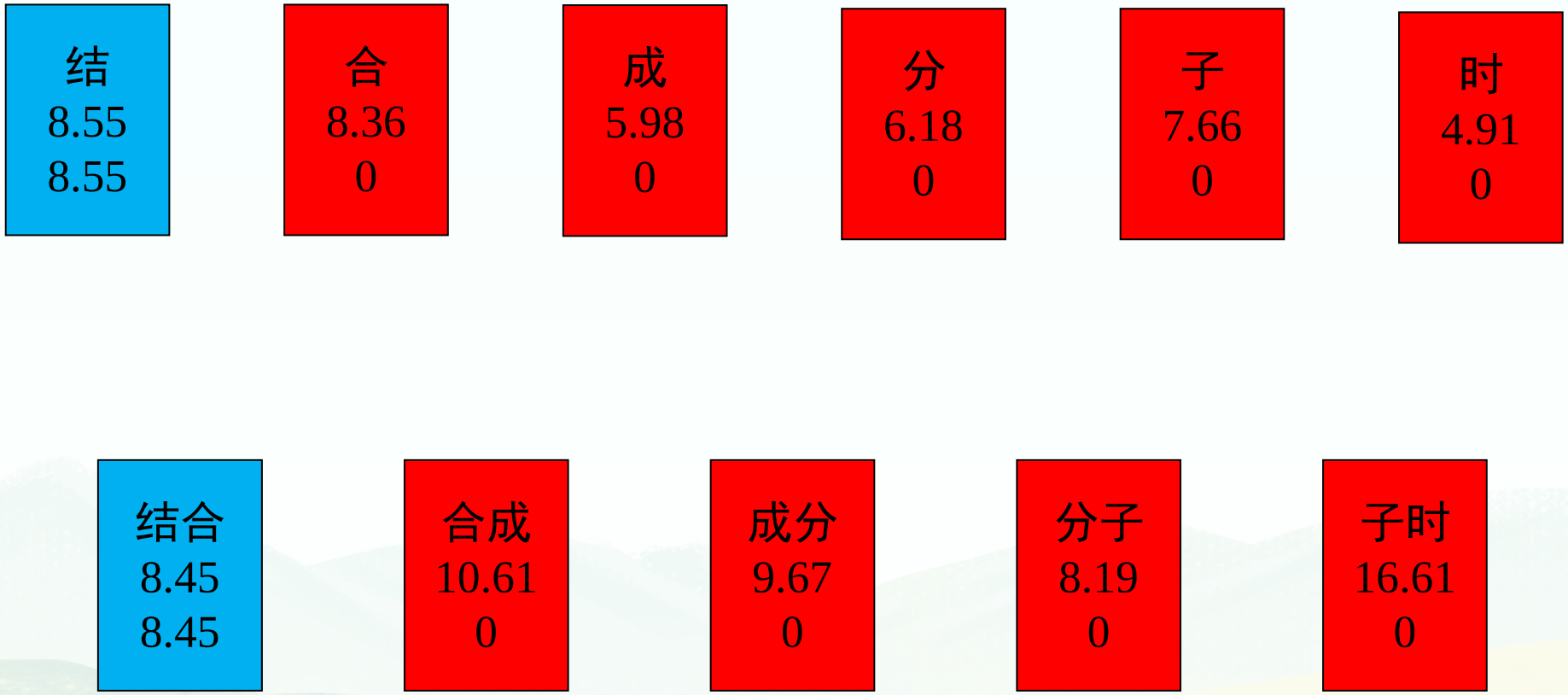
动态规范算法示例

待分词句子：结合成分子时

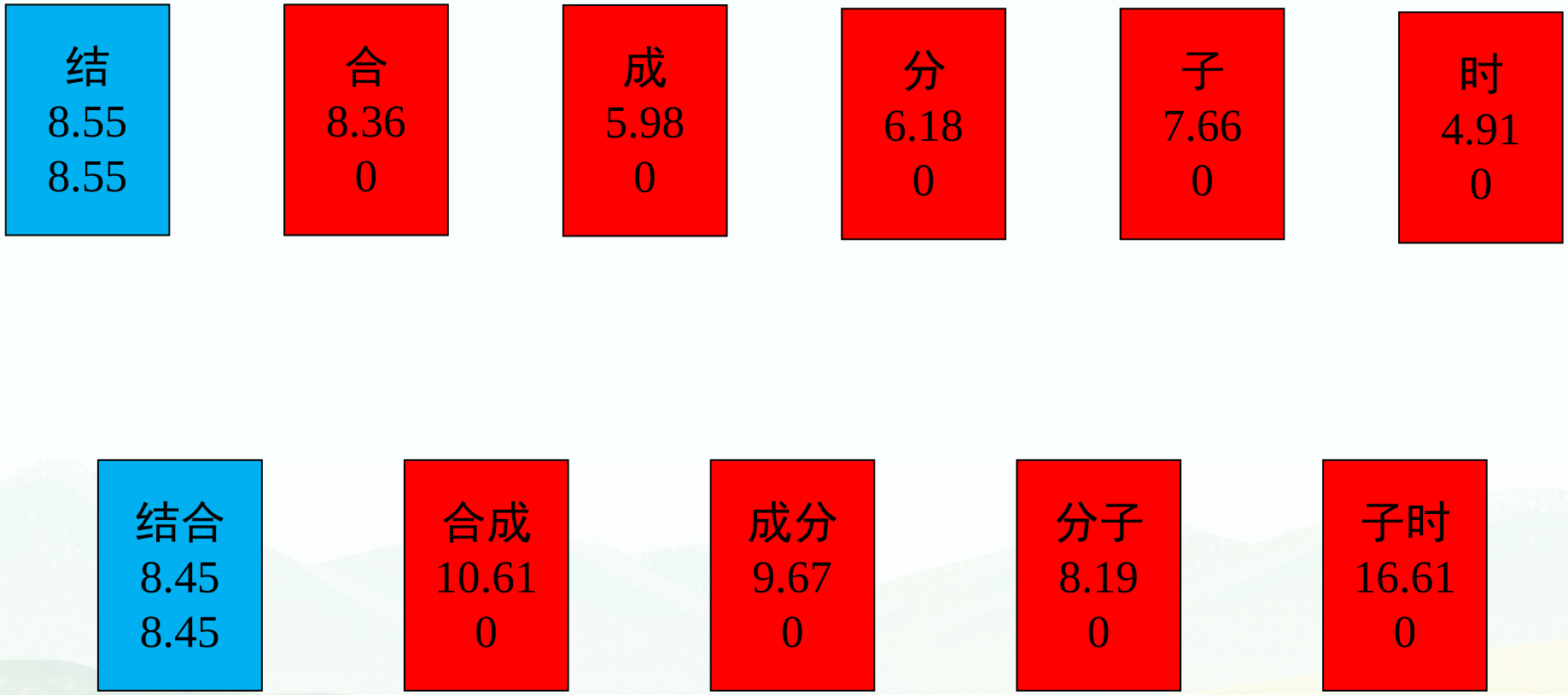
候选词及其费用（概率的-log）



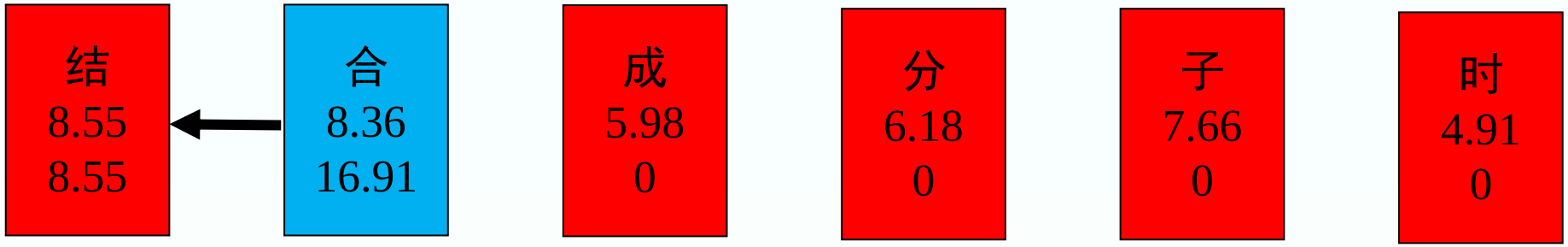
动态规范算法示例


结
8.55
8.55
合
8.36
0
成
5.98
0
分
6.18
0
子
7.66
0
时
4.91
0
结合
8.45
8.45
合成
10.61
0
成分
9.67
0
分子
8.19
0
子时
16.61
0

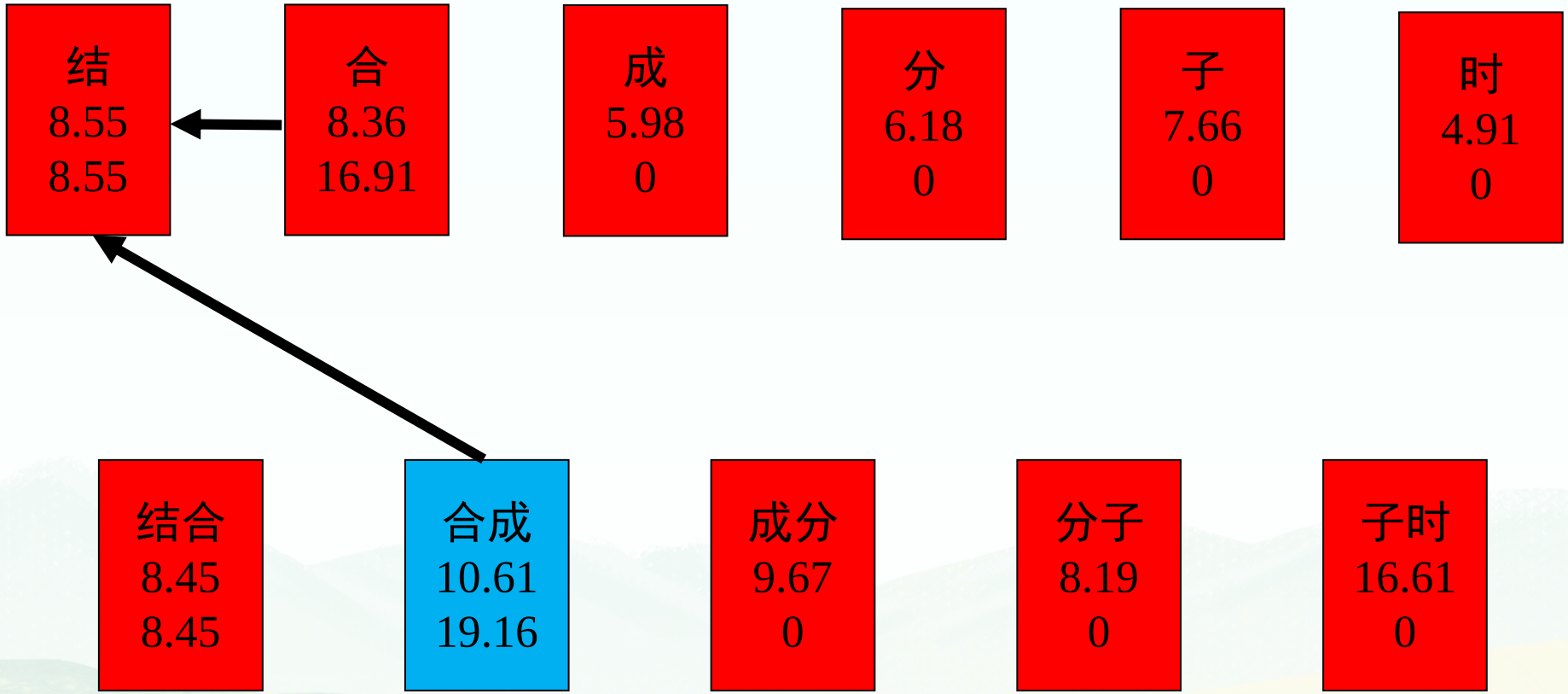
动态规范算法示例


结
8.55
8.55
合
8.36
0
成
5.98
0
分
6.18
0
子
7.66
0
时
4.91
0
结合
8.45
8.45
合成
10.61
0
成分
9.67
0
分子
8.19
0
子时
16.61
0

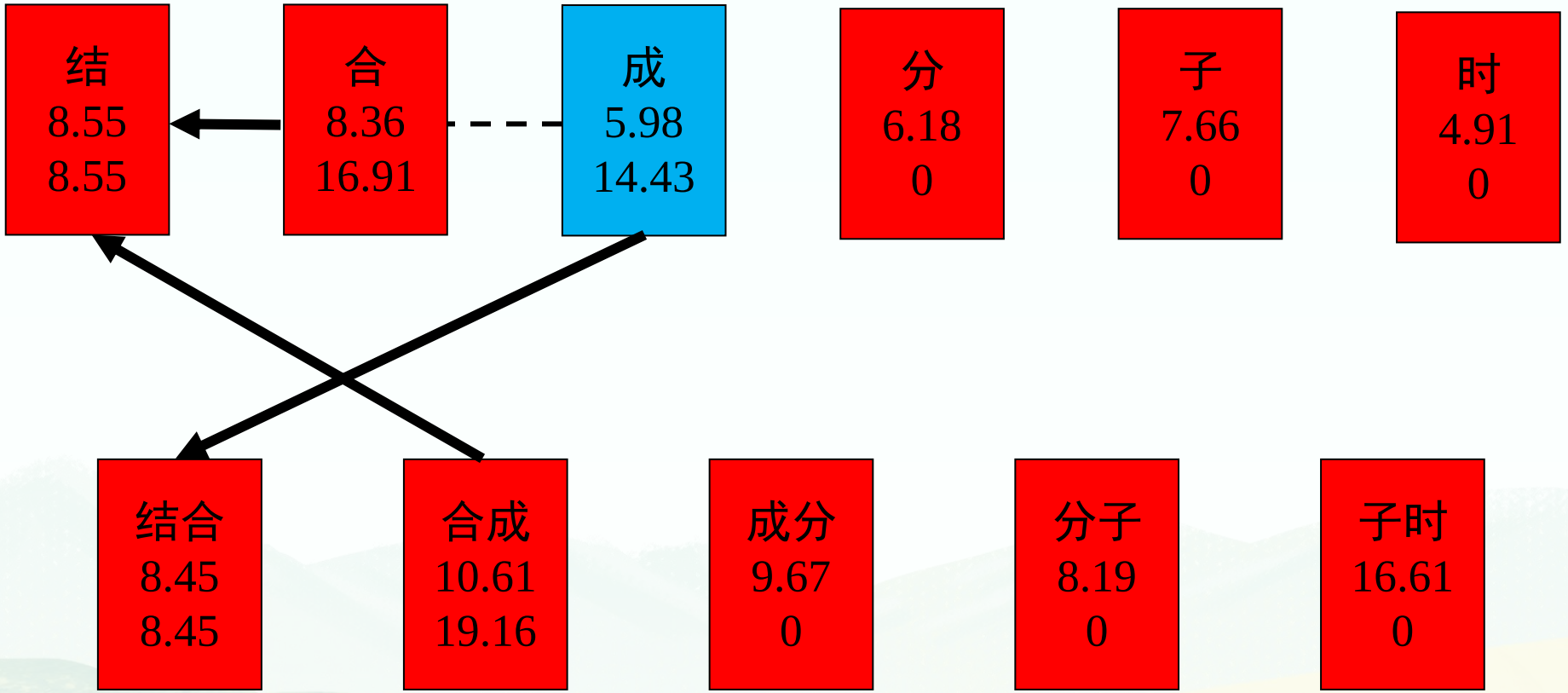
动态规范算法示例



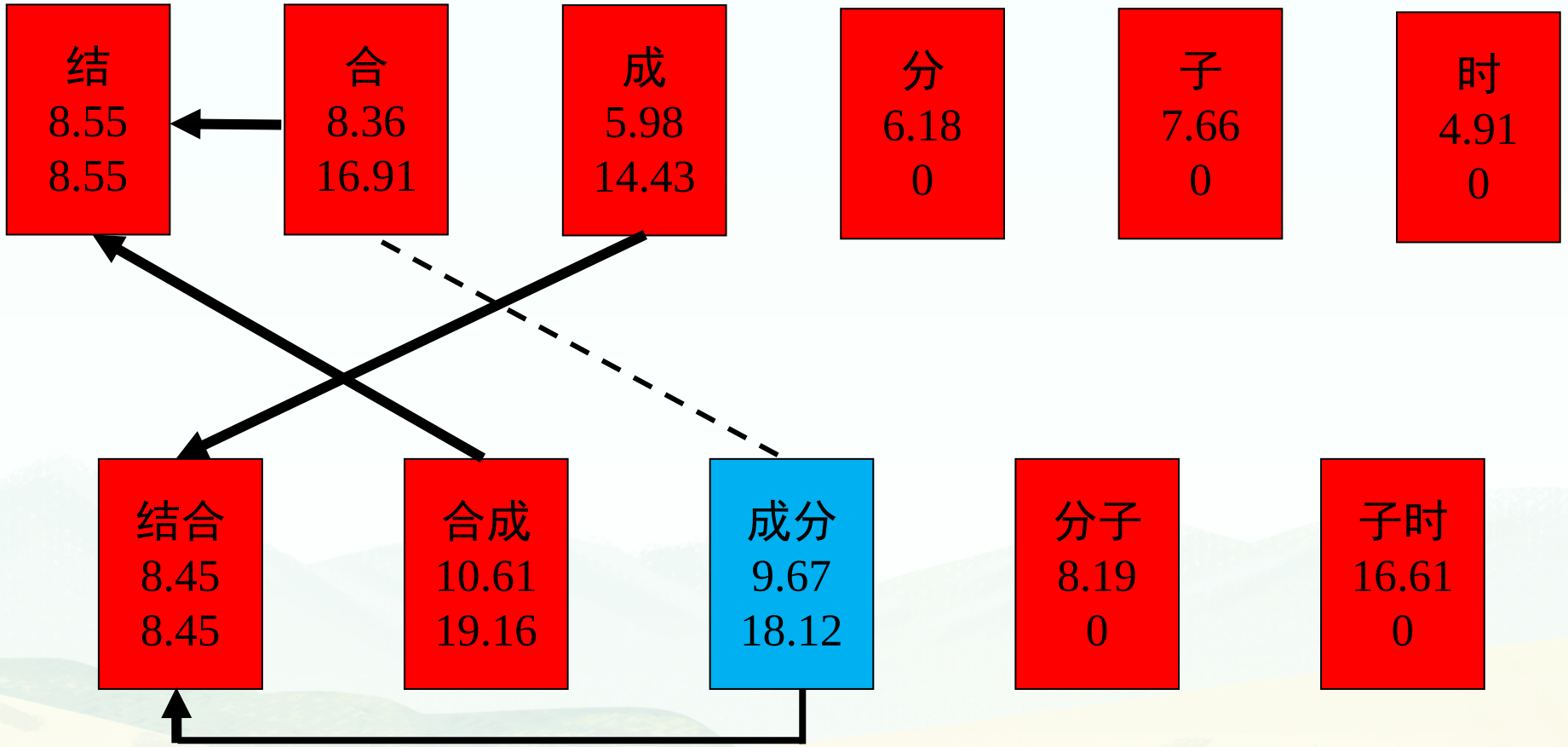
动态规范算法示例



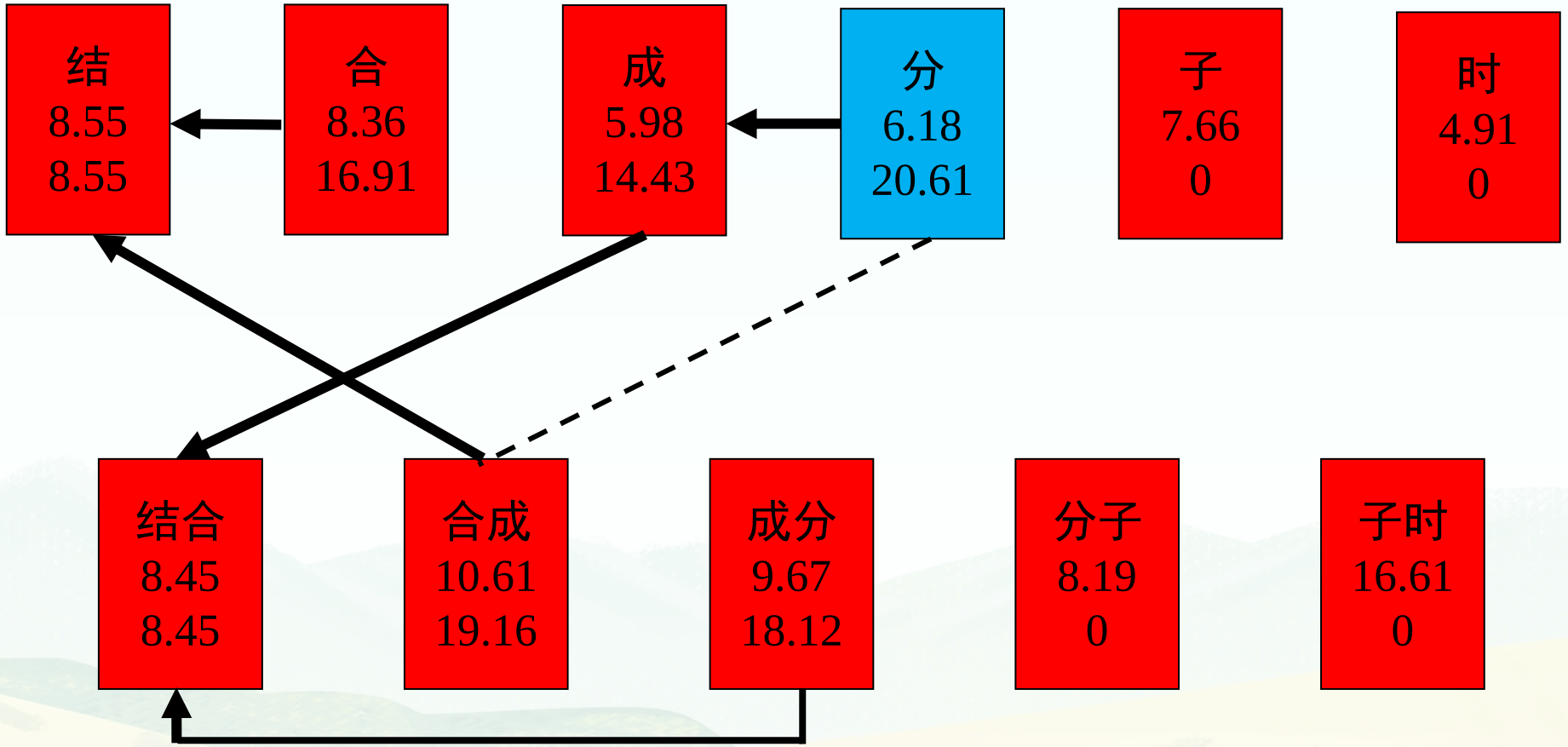
动态规范算法示例



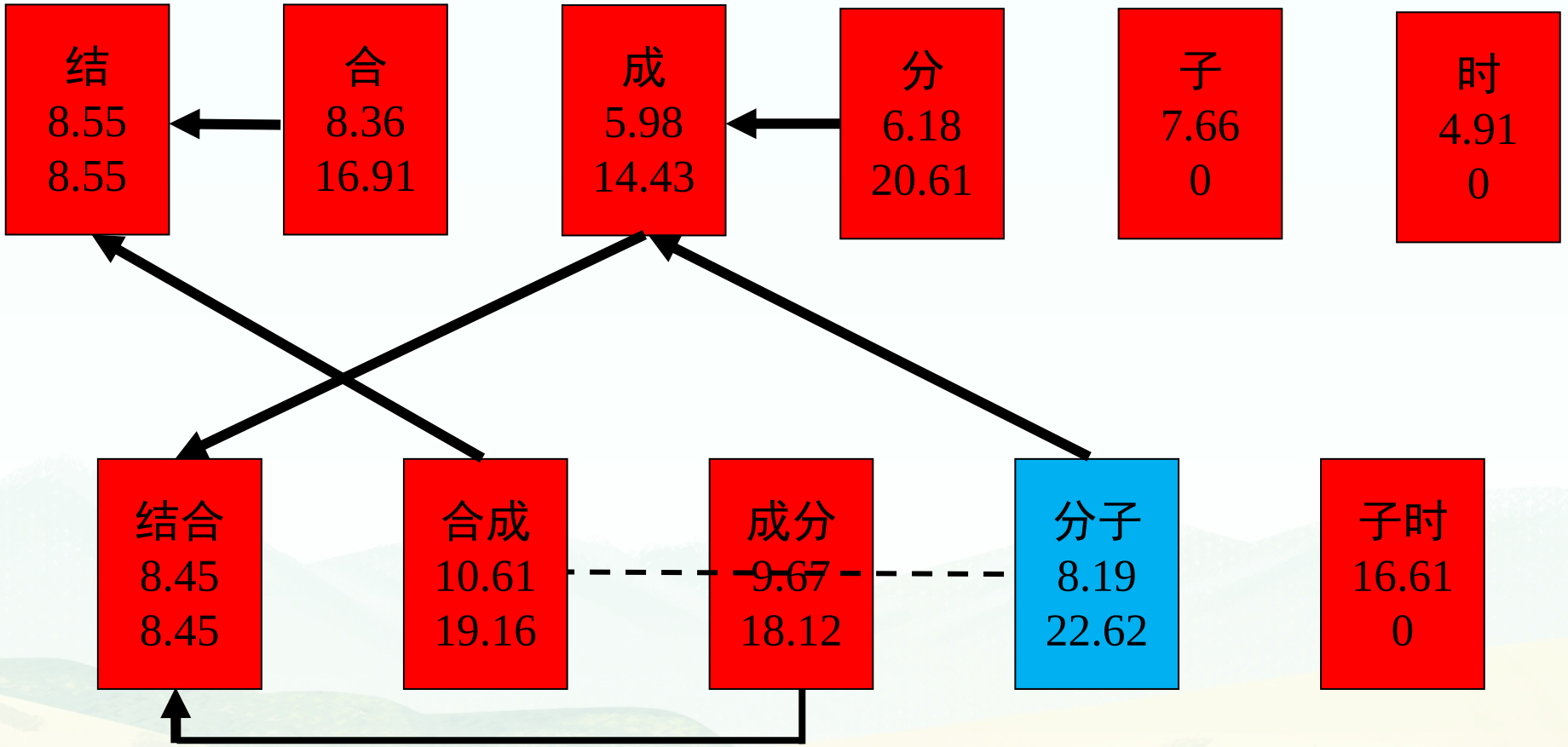
动态规范算法示例



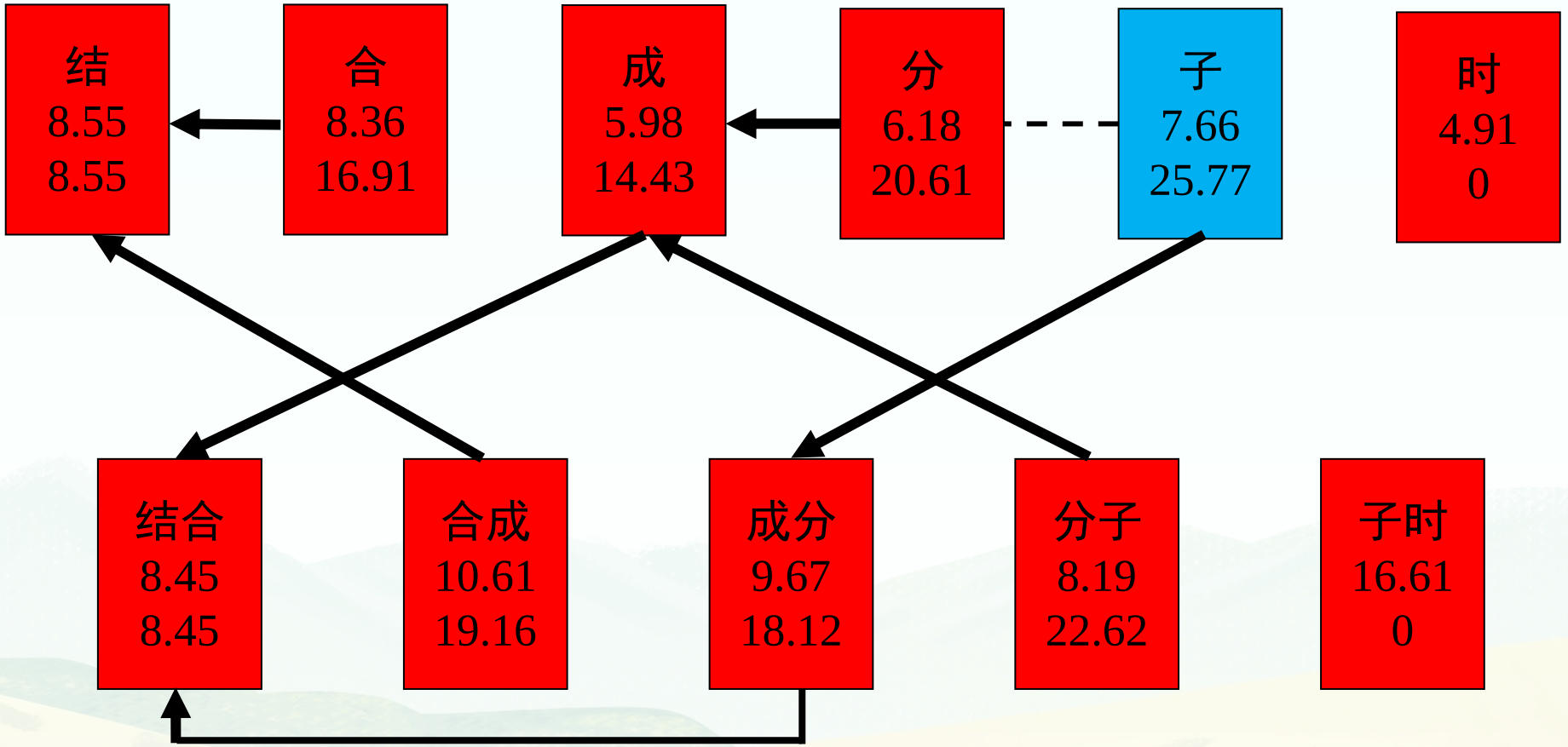
动态规范算法示例



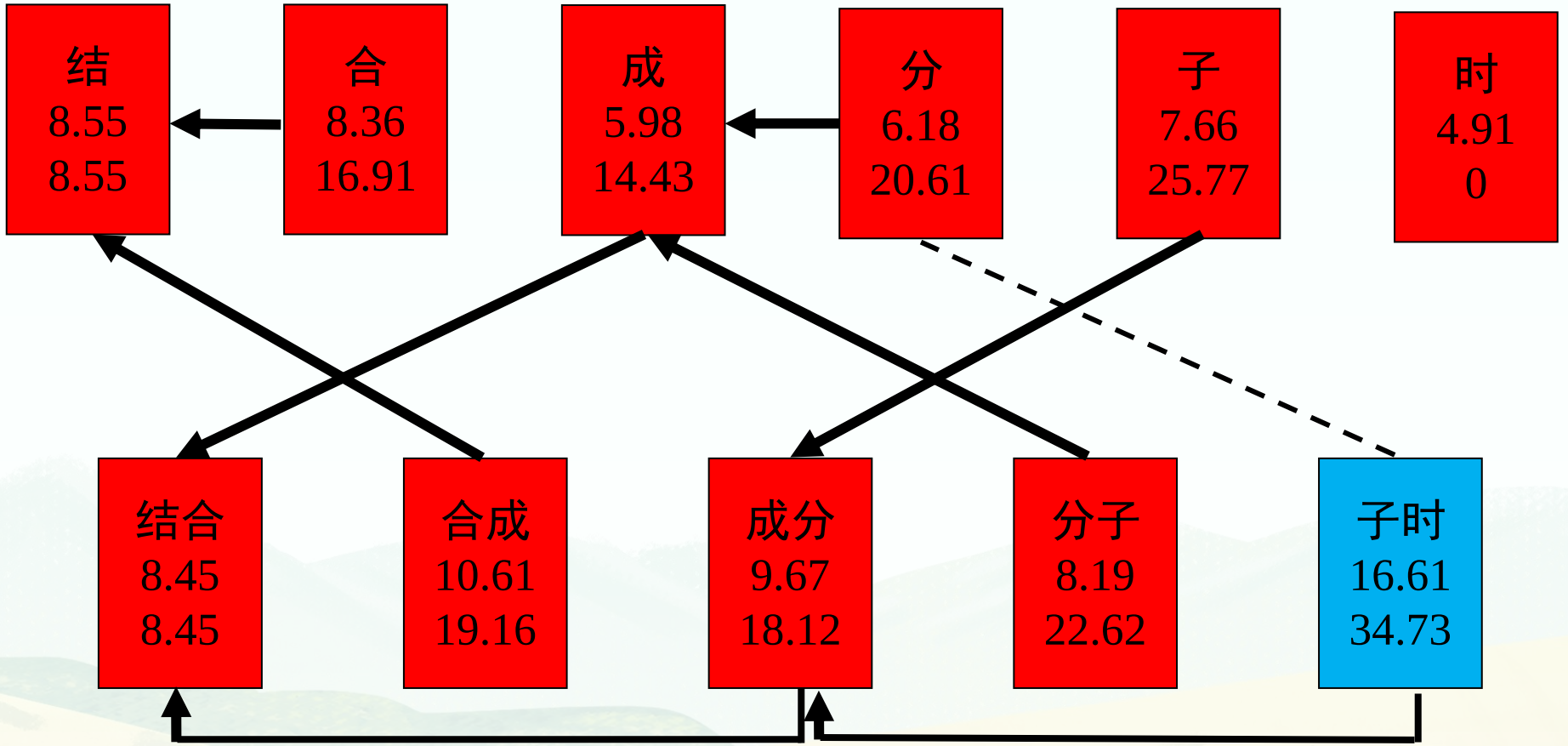
动态规范算法示例



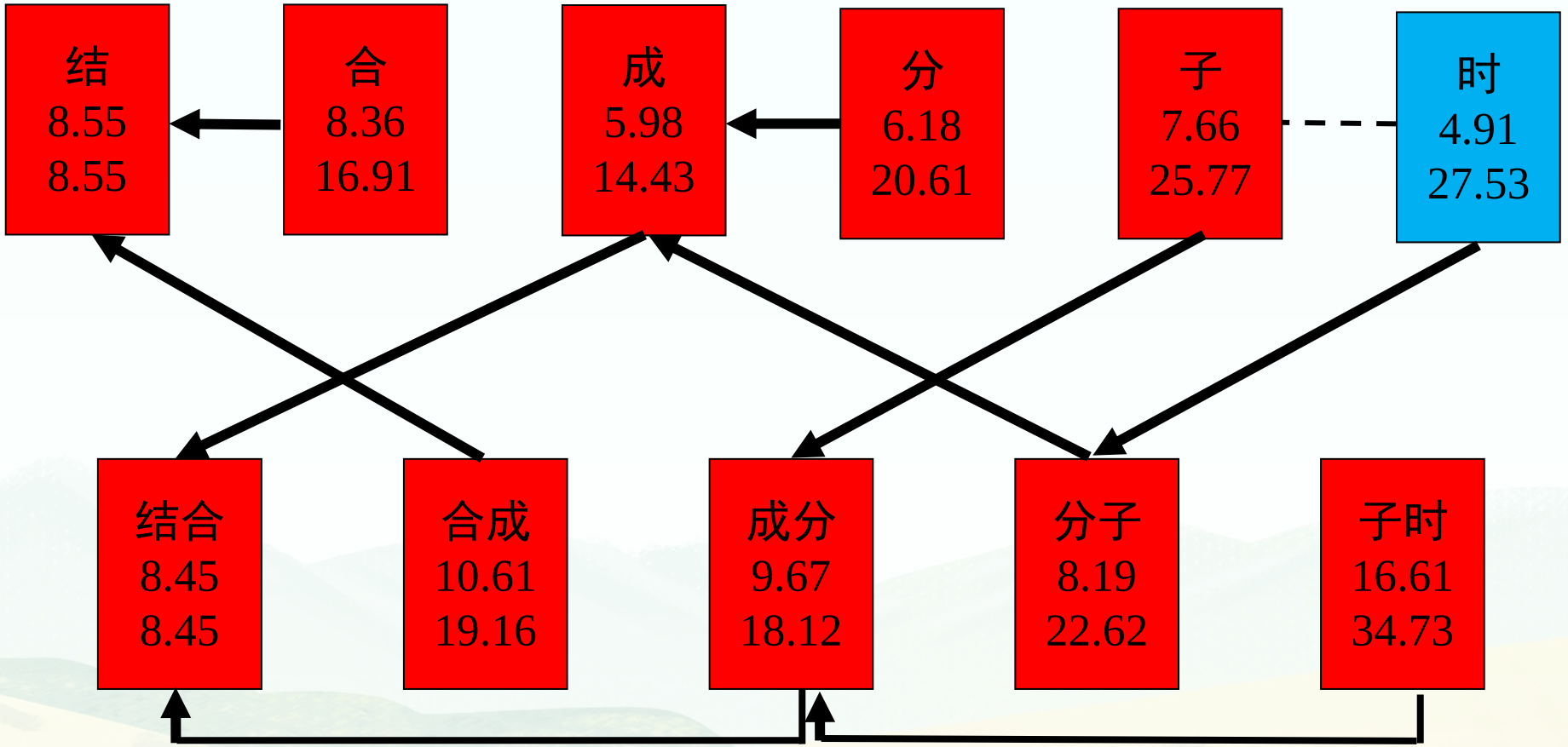
动态规范算法示例



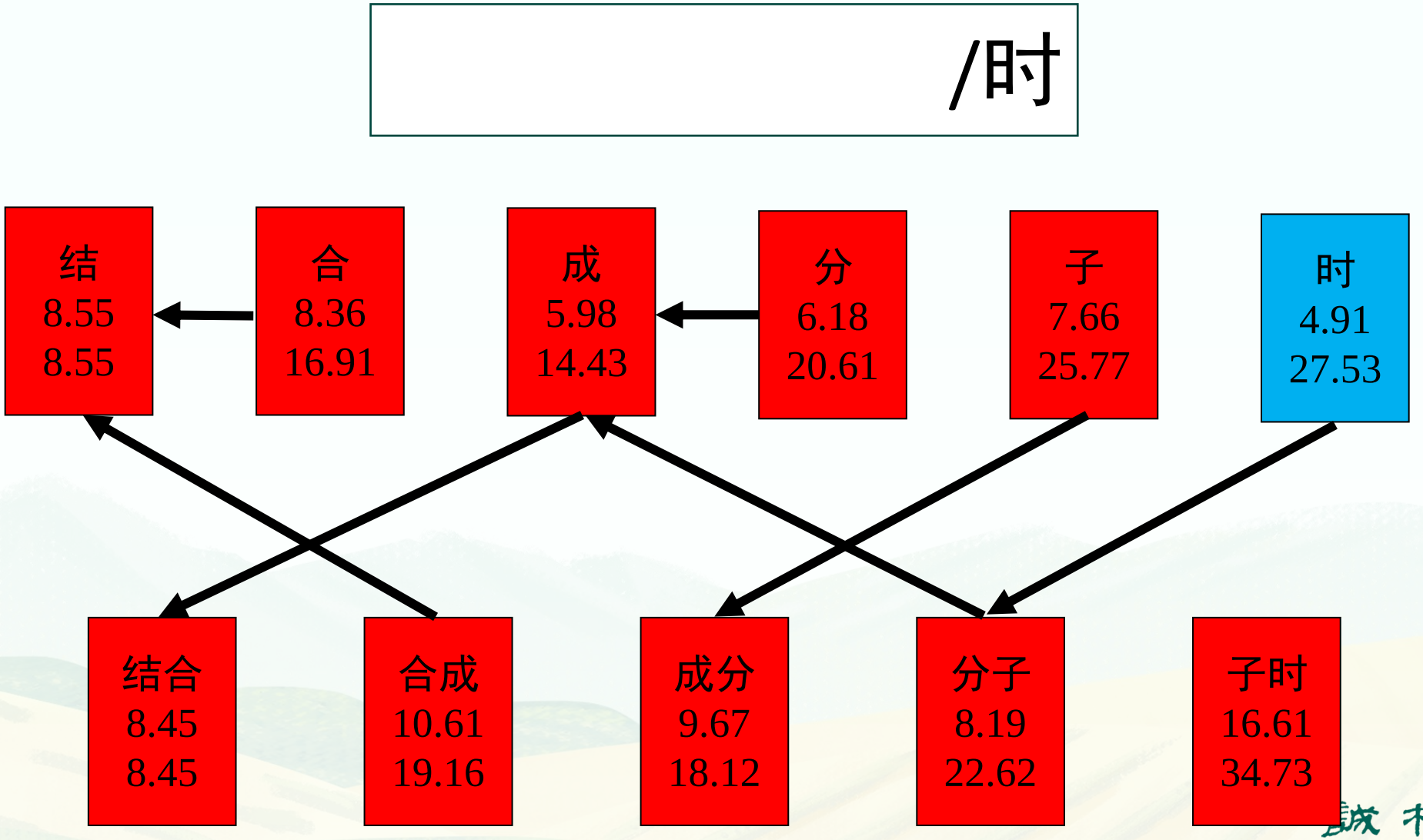
动态规范算法示例



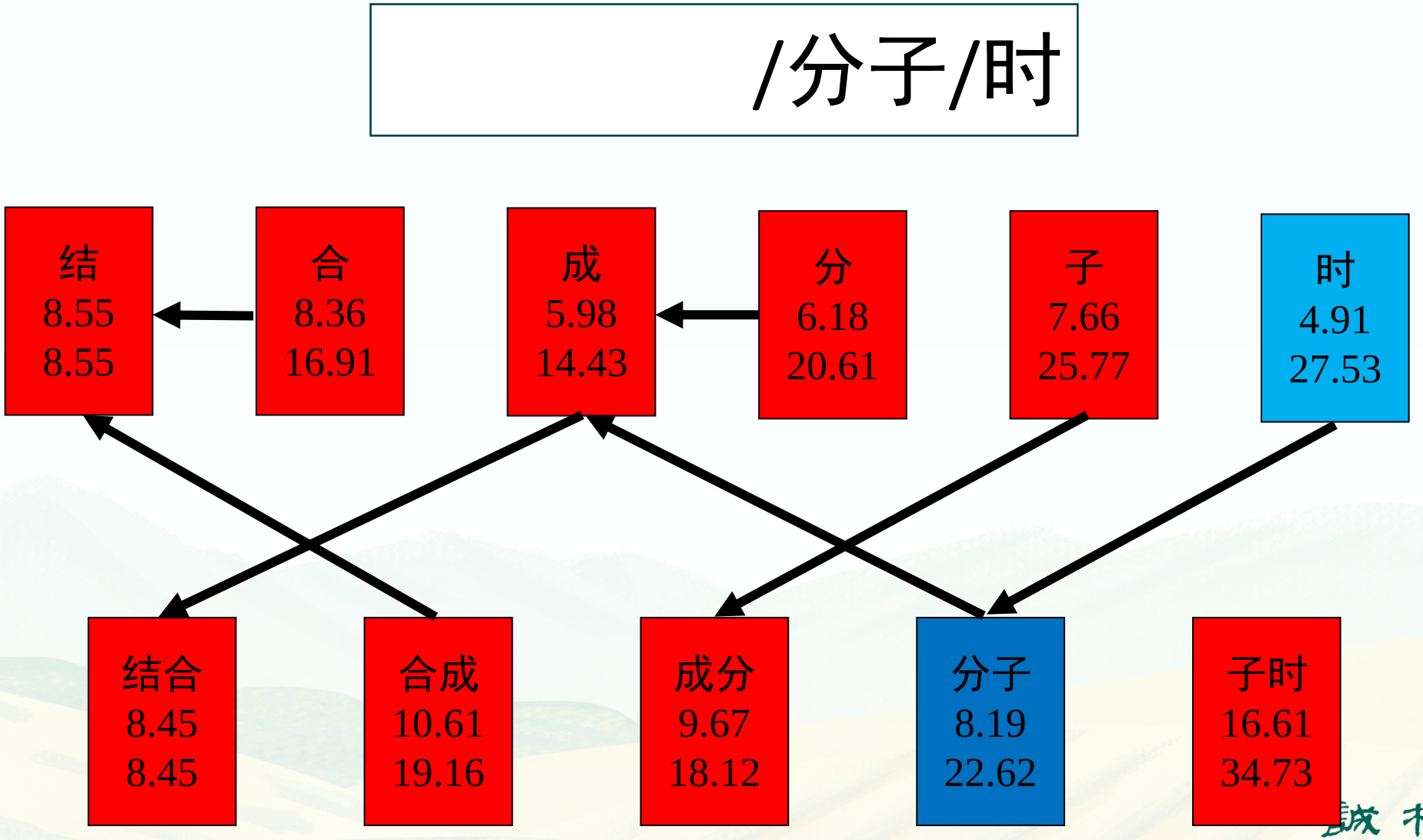
动态规范算法示例



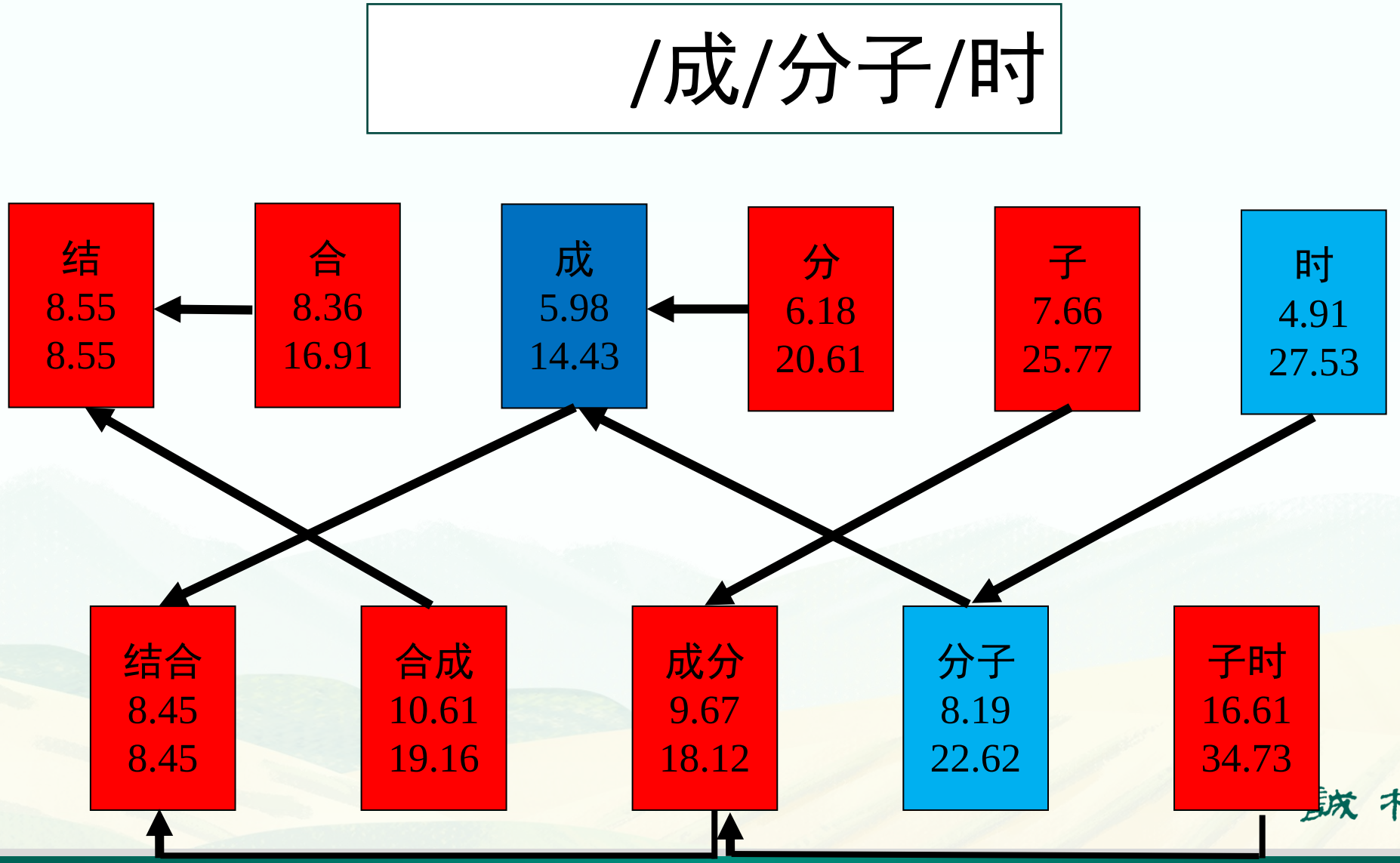
动态规范算法示例


/时
结
8.55
8.55
合
8.36
16.91
成
5.98
14.43
分
6.18
20.61
子
7.66
25.77
时
4.91
27.53
结合
8.45
8.45
合成
10.61
19.16
成分
9.67
18.12
分子
8.19
22.62
子时
16.61
34.73

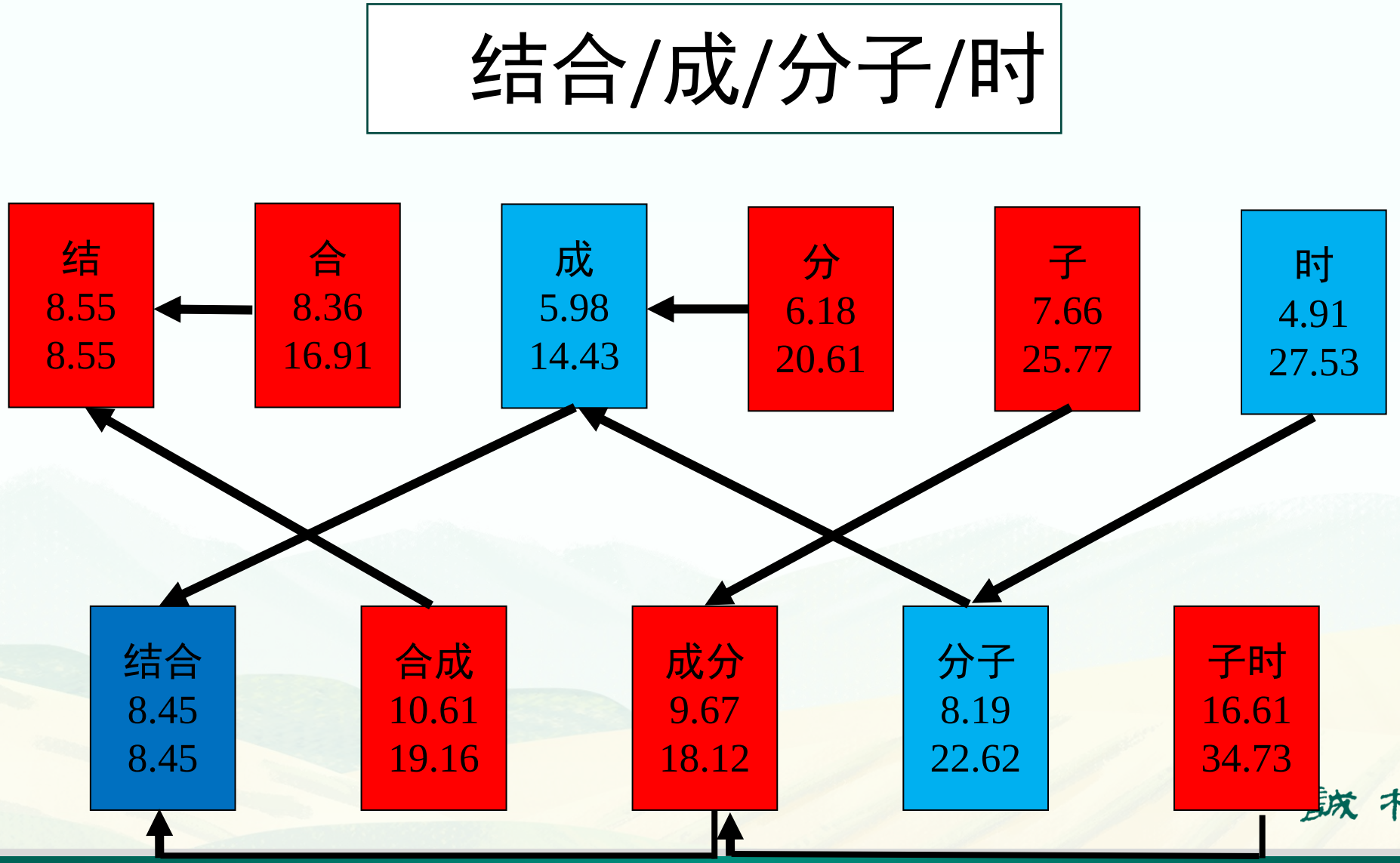
动态规范算法示例


/分子/时
结
8.55
8.55
合
8.36
16.91
成
5.98
14.43
分
6.18
20.61
子
7.66
25.77
时
4.91
27.53
结合
8.45
8.45
合成
10.61
19.16
成分
9.67
18.12
分子
8.19
22.62
子时
16.61
34.73

动态规范算法示例


/成/分子/时
结
8.55
8.55
合
8.36
16.91
成
5.98
14.43
分
6.18
20.61
子
7.66
25.77
时
4.91
27.53
结合
8.45
8.45
合成
10.61
19.16
成分
9.67
18.12
分子
8.19
22.62
子时
16.61
34.73

动态规范算法示例


结合/成/分子/时
结
8.55
8.55
合
8.36
16.91
成
5.98
14.43
分
6.18
20.61
子
7.66
25.77
时
4.91
27.53
结合
8.45
8.45
合成
10.61
19.16
成分
9.67
18.12
分子
8.19
22.62
子时
16.61
34.73

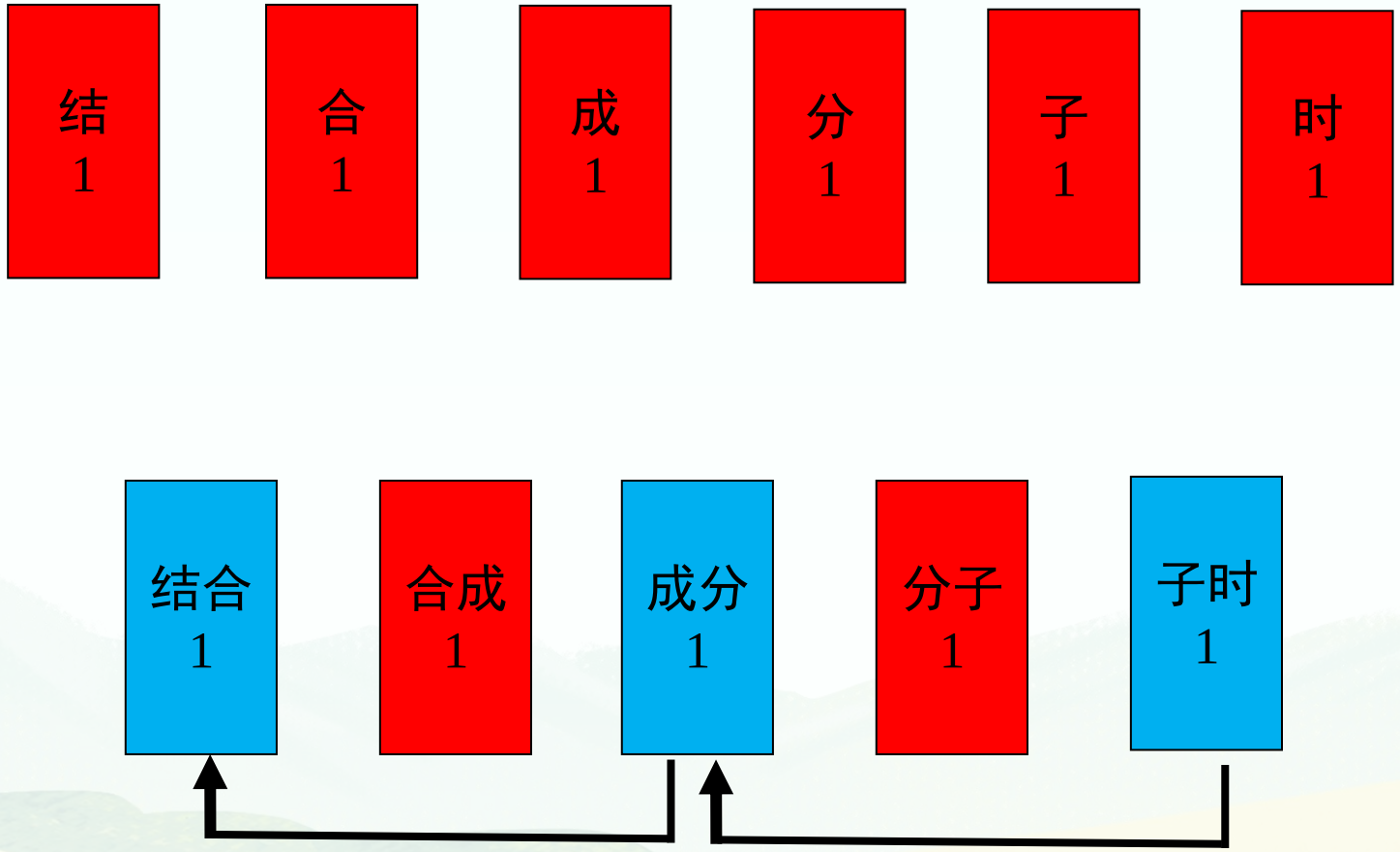
朴素算法与动态规划算法的比较

- 朴素算法：计算每一条路径上词的概率乘积，该例共需要做45次乘法。然后比较这13个概率乘积，需要做12次比较。
- 动态规划算法：该例“合”只有一个前驱，需做1次乘法，以后4个位置都是有两个前驱，分别需要做2次乘法和1次比较。选最优路径，需要做1次比较。总共是9次乘法和5次比较。
- 模型是用来模拟事物规律的，应在可计算前提下尽可能近似。算法则是用来解决计算效率问题的。

一元模型分词的评价

- 若每词概率相等，则退化为最大匹配法
- 分词精度
  - 一般在90%左右，比最大匹配法高
- 切分歧义
  - 没有利用上下文信息，对交集型歧义字串采取千篇一律的切分方式
  - 对于组合型歧义的消解基本无效
  - 对于交集型歧义：
    - 伪歧义消解效果好
    - 真歧义消解效果差
- 可尝试利用词的转移概率（二元模型）

一元模型分词的评价



一元模型分词

- 一个字串可对应于若干个词串。
- 不仅需要考虑词的概率
- 还要考虑每个词的bigram的概率
- 词串的概率为每个词的bigram之积
- P(w1,w2,...,wn)=P(w1)P(w2|w1)...P(wn|wn-1)

$$P(w_1, w_2, \ldots, w_n) = P(w_1)\prod_{i=2}^{n} P(w_i \mid w_{i-1})$$

一元模型的参数

结
8.55

合
8.36

成
5.98

分
6.18

子
7.66

时
4.91

结合
8.45

合成
10.61

成分
9.67

分子
8.19

子时
16.61

一元模型的参数

结
8.55

合|结
?

成|合
?

分|成
?

子|分
?

时|子
?

结合
8.45

成|结合
?

分|合成
?

子|成分
?

时|分子
?

合成|结
?

成分|合
?

分子|成
?

子时|分
?

二元模型参数求解

- 条件概率P(B|A)=P(AB)/P(A)
- 条件概率费用
  - –logP(B|A)=-log(P(AB)/P(A))
- 注意，这里的AB应该是词串，不能等同于字符串
  - 比如P(合|结)=P(结/合)/P(结)
  - P(结/合)≠P(结合)

隐马尔可夫模型

- 二元模型
  - 初始状态概率
  - 状态转移概率
- 隐马尔科夫模型的变化
  - 隐藏状态序列（观察不到）
  - 增加观察序列
  - 增加观察状态与隐藏状态之间条件概率
    - 隐藏状态条件下，观察状态的概率
    - 形象地看，可以叫做发射概率
- 隐马尔可夫模型的三组参数
  - 初始状态概率矩阵（π）
  - 状态转移概率矩阵（A）
  - 发射概率矩阵（B）

隐马尔可夫模型的公式

$$T^* = \underset{T}{argmax} \prod_{i=1}^{n} P(W_i \mid T_i) P(T_i \mid T_{i-1})$$

- Ti表示隐藏状态，Wi表示观察状态
- P(T1):初始隐藏状态概率
- P(Wi|Ti):发射概率
- P(Ti|Ti-1):隐藏状态转移概率

古汉语语料库的分词策略

- 先秦汉语语料的特点
  - 资源
    - 无训练语料
    - 无分词底表
  - 语言特点
    - 与现汉的用字、用词、语法差异大
    - 单字词比例高
- 两种策略
  - 人工制作训练语料或分词底表，采用机器学习的方式来加工（先学后用）
  - 机器辅助制作词表和训练语料（人机交互）
- 综合：先人机交互，再机器学习

分词的评测

- SIGHAN(Special Interest Group of HAN)
  - 正确率、召回率、F1值
  - OOV的召回率
- 更深入地
  - 交集型歧义消解能力
  - 组合型歧义消解能力
  - 评测语料的分词一致性如何
  - 按句子计算

南京農業大學
NANJING AGRICULTURAL UNIVERSITY

自动分词在数字人文研究中的应用背景

汉语自动分词是数字人文研究的重要前提，是深度挖掘经典文献和深入研究传统文化的必要根基。作为最小表意单元，汉语词汇间并不具有天然分隔。面对海量现代与古代汉语文本，完全依赖手工分词不仅工作量巨大，还难以保证分词结果的一致性与规范性。因此，需要借助现代计算机信息技术，自动化完成汉语词汇切分。

基于规则的自动匹配分词

汉语自动分词主要可以分为基于规则的自动匹配分词和基于概率统计分词两类方法。

◆前者通常通过人工标注或引入外部词典信息构建领域词汇表，融合停用词表获得分词底表。例如，黄建年 通过N元语法（N-gram）和词典分词技术在农业古籍上实现了自动分词，徐润华和陈小荷 构建了注疏词表，采用最大匹配分词算法对《左传》进行了分词。但是，由于基于规则匹配的方法通常无法识别未登录词，因此当前大多采用基于概率统计的机器学习或深度学习方法进行汉语自动分词。虽然目前已经出现了诸如NLPIR 、Jieba 、HanLP 、NLTK 等中文分词工具可供直接使用，但由于上述软件内置的分词模型大多基于通用语料训练，因此难以在依赖语言学、历史学、文献学等领域数据和知识的数字人文研究中使用。

基于概率统计分词

采用统计学习模型的方法能够根据语料库先验概率与条件概率分布自动判断词汇边界。其分词过程大致如下：首先，由相关专业标注人员进行手工词汇切分，经校验后形成精标数据集。其次，基于人工标注语料构建训练集，制定不同人工特征模板。最后，采用序列标注方法训练得到最佳分词模型，并完成对全部未标注语料的自动分词。

基于概率统计分词

条件随机场模型（CRF）在汉语自动分词中的应用最为广泛。石民等 编制了面向计算机信息处理的《古代汉语分词标注规范》，采用条件随机场模型在《左传》上进行了自动分词实验，发现字符分类、二元同现、声调等特征能够提高分词性能。留金腾等 采用CRF自动标注和人工校验的方式构建了《淮南子》全文分词及词性标注语料库，通过领域适应方法和语言学特征融合，仅需要少量人工标注的古文数据即可快速获取高质量分词模型。黄水清等 基于《春秋经传注疏引书引得》构建领域词表，并作为外部词汇知识融入CRF特征学习过程，在《左转》和《晏子春秋》上提高了自动分词的准确率，为先秦典籍的自动分词提供了新的视角。欧阳剑 基于二元文法（Bigram）和CRF模型完成了《左传》自动分词，并通过可视化数据挖掘的方式搭建了服务于语言学、文献学、历史学的古籍统计分析平台。王晓玉和李斌 将词典标记信息和字符分类特征加入 CRF的特征模板，有效提升了在史书、佛经等中古语料上的分词性能。

基于概率统计分词

基于统计学习模型的自动分词方法对有标记数据集的规模和计算机硬件资源的要求相对较低，适合于面向小样本监督数据集和移动平台的自动分词模型训练与应用。但是要想构建高性能的分词模型，需要具备相关领域知识的专家通过特征工程统计多维语言学特征，构建复杂的机器学习特征模板，因此实现与普及的门槛较高。

基于概率统计分词

深度学习模型无需人工选择待分词文本特征，神经网络架构能够自行在大规模标记数据集上提取丰富的语义与关联特征。当前较为主流的研究方法是采用Word2Vec词嵌入工具对文本进行向量化表示，继而基于RNN、Bi-LSTM、Transformer等深度神经网络架构完成自动分词。王莉军等 基于Bi-LSTM-CRF模型对中医领域文本进行了自动分词，其效果较Jieba等通用分词工具有大幅提升。程宁等 则基于一体化分词与词性标注的思想，采用Bi-LSTM-CRF模型完成了不同历史时期古籍的自动分词，该模型分词准确率高于IDCNN模型，且优于单独分词的效果。

基于概率统计分词

基于Transformer架构的预训练语言模型具有更强的语义表征能力，尤其是面向数字人文领域相关任务继续训练的模型可以更加充分的学习到特定领域文本的句法与文法规则。采用小样本标注语料进行微调，即可利用BERT、RoBERTa、SikuBERT等进行全文分词。张琪等 基于BERT模型在先秦典籍语料上训练了分词词性一体化标注模型，其分词F值超越了机器学习模型CRF和神经网络模型GRU-CRF，并利用该模型对《史记》文本进行了自动标注与知识挖掘。王东波等 通过领域适应性训练的方式，在《四库全书》全文语料上进一步预训练了通用BERT，构建并发布了面向古文自动处理的SikuBERT模型。刘畅等 利用该模型在无标点的《汉书》《三国志》等古籍语料上进行了自动分词对比实验，分词表现优于Bi-LSTM-CRF和现代汉语预训练的BERT模型。

基于概率统计分词

基于深度学习的分词方法在相同数据集上往往能够取得超越传统机器学习模型的分词表现。这一方面得益于更加复杂且深度的神经网络结构能够学习到更多显式和隐式的词法与句法特征，另一方面用于支撑模型训练的大规模标注数据集包含较为全面的自然语言知识，使得模型在开放测试中具有较强的泛化能力。尤其是近几年火热的预训练语言模型，在预训练阶段以无监督的方式充分学习海量真实文本的语言学特征与词汇、句子关联，面向下游任务仅需通过迁移学习与领域微调的方式即可取得优异的分词性能。

南京農業大學
NANJING AGRICULTURAL UNIVERSITY

语言信息处理中的统计方法

为什么要学统计?

语言的规律性和随机性

◆语言的使用、语言的发展都是有规律的。

◆但是，自然语言远不是一个经过事先精心规划的系统，我们难以用一套规则去准确地预测真实语料中所出现的各种变异，也就是说，这些变异有相当的随机性。

◆例如在汉字识别和音字转换时，很难制定完整有效的规则从候选字矩阵中选择一条正确的路径。

◆当我们把握不住语言的规律时，承认语言的随机性并求助于概率统计，是明智和务实的。

概率论和统计学

- 面对语言现象的随机性，需要有一种研究随机现象的工具。
- 概率论研究随机现象中有关事件的规律性。
- 统计学研究如何以有效的方式收集、整理和分析受到随机性影响的数据，从而对所考察的问题作出统计推断。这种统计推断是以概率论的理论为基础的。

语言统计的两种取向

- 一种是对样本进行数据分析，希望从中推断出关于总体的结论，例如比较两种语言教学方法的效果，看看差异是否显著。这种取向常见于语言教学研究，样本规模通常较小。
- 另一种是从样本（训练语料）中获取语言模型的参数，然后用这个模型来分析总体中的其他数据（测试语料）。这种取向常见于自然语言处理的研究，样本规模通常很大。

概率（probability）

- 设样本空间Ω中共有n个样本点，事件A有m个，则事件A的概率为： P(A) ＝ m / n
- 例：一个语料库有835万词次，其中单词“为”出现3万次。若从该语料库中随机挑选一个词，这个词恰好是“为”的概率是多少？
- 解：语料库规模足够大，可用相对频率来近似概率。此时Ω中共有835万个样本点，事件A（单词是“为”）有3万个样本点，因此P(A)为3/835.

概率的性质

- 非负性：$P(A) \geq 0$
- 规范性：$P(\Omega) = 1$　　（Ω表示全部基本事件）
- 可加性：对于无穷多个事件A1,A2,...，如果事件两两互不相容（相互独立），则 $P(\cup Ai)=\Sigma P(Ai)$
- 例：如果“为”字仅有两种读音，读wei4的概率为0.6，那么“为”字读音wei2的概率是多少？
- 解：利用概率的规范性和可加性，“为”字读音wei2的概率是 1－0.6＝0.4

联合概率（Joint Probability）

- P(AB)是事件A和B都发生的概率，叫联合概率。
- P(AB) = P(BA).
- 例：求某字写做“为”且读音是wei4的概率。
- 解：这里P(A)是“为”字的概率，P(B)是读音wei4的概率。求P(AB)的方法之一是从语料库中统计“为”读wei4的次数，用它除以该语料库所有字符的出现次数。
- 已知P(A)和P(B)，是否可以据此求出P(AB)呢？

条件概率（conditional probability）

- 已知事件B发生的条件下事件A的概率叫做A的条件概率：

- 当 P(B) > 0 $$P(A|B)=\frac{P(AB)}{P(B)}$$

- 由此可知 P(AB) = P(A|B) P(B)= P(B|A) P(A)

转移概率（transitive probability）

- 转移概率是从一个状态转移到另一个状态的概率，亦即事件先后发生的条件概率。
- P(W2=“的” | W1=“绿油油”)，两事件先后发生，因此既是条件概率，也是转移概率。
- P(Wi=“编辑” | Ti=名词)，两事件同时发生，因此只是条件概率，不是转移概率。
- 转移前后的两个状态可以相同，例如，句子中连续出现两个名词，就是从名词转移到名词。
- P(w1)P(w2|w1)P(w3|w1w2)中，第一项不是转移概率，第二项是二元转移概率，第三项是三元转移概率。

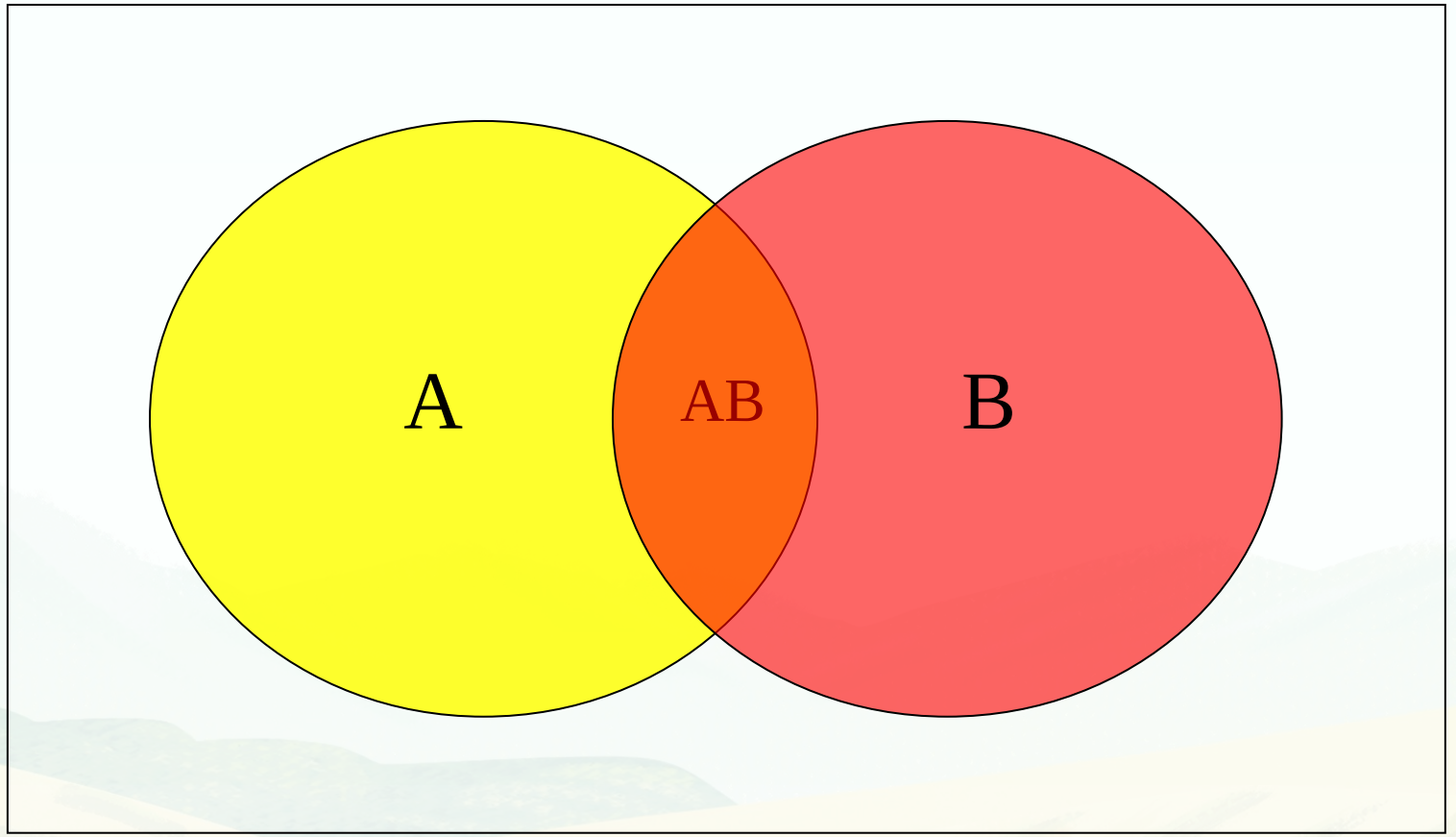
条件概率的计算

- 例：求“为”字读音是wei4的概率。
- 解：已知所考察的字是“为”，在这种情况下求它读音是wei4的概率。因此这是求条件概率。
- P(Z读音wei4 | Z写做“为”)
  ＝ P(Z写做“为”且Z读音wei4) / P(Z写做“为”)
  ＝ (“为”读wei4的次数 / 所有字符的出现次数)
  ÷ (“为”的出现次数/所有字符的出现次数)
  ＝ “为”读wei4的次数 / “为”的出现次数

联合概率与条件概率

- 已知条件概率和无条件概率，可以求联合概率：
- $P(AB) = P(A)\,P(B|A) = P(B)\,P(A|B)$
- 仅知两个无条件概率，无法求联合概率。除非知道 $P(B|A)=P(B)$，（A的发生对于B的发生毫无影响，既不促发也不抑制），才有 $P(AB)=P(A)P(B)$。

概率、联合概率、条件概率的关系



概率、联合概率、条件概率的关系

- 证明以下不等式：
- $P(A) \geq P(AB)$
- 一个样本点若属于AB，必然也属于A，但反之不然。因此A中样本点必不少于AB中的样本点。何时等式成立？
- $P(A|B) \geq P(AB)$
- 因为$P(AB)=P(A|B)P(B)$，且$0 \leq P(B) \leq 1$. 何时等式成立？
- P(A) 与 P(A|B) 孰大孰小？

先验概率和后验概率

- 先验概率（prior probability）：在缺乏证据的情况下，人们对于一个事件的可能性大小的描述。例如，随意抽取文本中的一个词，该词是名词的概率 P(名词) 就是先验概率。
- 后验概率（posterior probability）：在已经得到某种证据的情况下，人们对一个事件的可能性大小的描述。例如，已知某词是名词，它带后缀的条件概率 P(带后缀|名词) 就是后验概率。

贝叶斯公式（Bayes’ Law）

- 如果事件A1，A2，．．．．，An两两互不相容，P(Ai)>0（i=1，2，．．．．，n），并且 $B \subseteq \bigcup_{i=1}^{n} A_i$ ，P(B)>0，那么，

$$P(A_i \mid B) = \frac{P(A_i B)}{P(B)} = \frac{P(A_i)P(B \mid A_i)}{\sum_{j=1}^{n} P(A_j)P(B \mid A_j)}$$

贝叶斯公式的用处

- 贝叶斯公式可以帮助我们根据先验概率和后验概率来决策（例如分类）。
- 分母 P(B) 与决策无关，故可简化为：
- $A^* = \text{argmax}_{Ai} P(A_i B)$
- $= \text{argmax}_{Ai} P(A_i) P(B|A_i)$
- 意思是，对于每个类别A_i，分别求出其先验概率和后验概率的乘积，其中使概率乘积最大的那个Ai就是我们所应选择的类别A*。

贝叶斯公式应用举例

- 已知某词带有后缀，它属于什么词类？
- 解：假定有n个词类（名词、动词、形容词等等）。
  根据贝叶斯公式，可以先求出：
- P(名词) P(带后缀|名词)
- P(动词) P(带后缀|动词)
- P(形容词) P(带后缀|形容词)
- ……
- 然后选择其中使概率乘积最大的那个词类。

概率的乘法定理

- P(AB) = P(A) P(B|A)
- P(ABC) = P(A) P(B|A) P(C|AB)
- 可推广到有限多个事件联合出现的情形：

  P(A1A2...An)

  =P(A1)P(A2|A1)P(A3|A1A2) ... P(An|A1A2...An-1)
- 以上就是概率的乘法定理（又称链规则）。
- 乘法定理是语言统计模型的基础。

概率乘法定理的应用

- 例如，P(Z是“为”且Z读音wei4)
  ＝ P(Z是“为”) P(Z读音wei4 | Z是“为”)
  ＝ P(Z读音wei4) P(Z是“为” | Z读音wei4)
- 又如，若w1w2w3是一个词串或标记串，则该串的概率为：

$$P(w1w2w3)=P(w1)\,P(w2|w1)\,P(w3|w1w2)$$
$$= P(w3)\,P(w2|w3)\,P(w1|w2w3)$$

信息论（Information Theory）

- 信息论是运用概率论与数理统计方法研究信息、信息熵、通信系统、数据传输、密码学、数据压缩等问题的应用数学学科。
- 香农被称为“信息论之父”。人们通常将香农于1948年10月发表于《贝尔系统技术学报》上的论文《A Mathematical Theory of Communication》作为现代信息论研究的开端。在该文中香农给出了信息熵的定义。

自信息（Self-Information）

- 随机变量X有若干个取值，它取值为x是一随机事件，该事件的概率的负对数叫做该事件的自信息：

  $$I(x) = - \log_2 P(x)$$

- 自信息可理解为成功猜测某事件所需最多次数。（猜测过程中，对方只回答是或否，不提供其他信息。）
- 例：若硬币朝上概率为0.5，则猜测次数为 -log0.5=2
- 例：若骰子点数为3的概率为1/6，则猜测次数为

  -log(1/6)=2.585
- 例：若“间”读阴平的概率是0.97，则猜测次数为

  -log(0.97)=0.044

熵（Entropy）

- 熵是随机变量的各相关事件的自信息的概率加权平均值：

$$
\begin{aligned}
H(X) &= \sum_{x \in X} P(x) I(x) \\
&= \sum_{x \in X} P(x)(-\log P(x)) \\
&= -\sum_{x \in X} P(x) \log P(x)
\end{aligned}
$$

熵的特点

- 熵是随机变量的不确定性的度量，熵越大表明不确定性越大，熵为0时表示完全可以确定。
- 概率分布越均匀，熵越大;在同样均匀的情况下，分布越广，可能性越多，熵越大。
- “自信息”的加权平均值，所以也是随机变量的平均信息量（bit）

熵的计算

例：掷硬币有两种结果，假定正面朝上和反面朝上的概率都是0.5；掷骰子有6种结果，假定每种结果的概率都是1/6。掷硬币的结果与掷骰子的结果这两个随机变量的熵孰大孰小？

解：H(掷硬币)＝－2(0.5log0.5)＝log2＝1.0

H(掷骰子)＝－6((1/6)log(1/6))＝log6=2.58

掷骰子的结果这个随机变量的熵较大。由此可见，随机变量的分布同样均匀时，分布越广的，熵越大。

熵的计算

例：据统计，“间”读阴平的概率是0.97，读去声的概率是0.03；“藏”读zang4的概率是0.56，读cang2的概率是0.44。“间”的读音和“藏”的读音这两个随机变量孰大孰小？

解：H(“间”的读音)

$= -0.97\log 0.97 - 0.03\log 0.03 = 0.19$

H(“藏”的读音)

$= -0.56\log 0.56 - 0.44\log 0.44 = 0.99$

“藏”的读音这个随机变量的熵较大。由此可见，概率分布广度相同时，分布越均匀的，熵越大。

熵的语言学意义

- 在一个语料库中，字串“色列”左边只能出现字符“以”，因此左字熵为0.0（没有不确定性），由此可认为“色列”左边是不自由的
- 字串“以色列”左右两边都能出现多种字符，并且没有哪种字符的概率特别高，因此左字熵和右字熵都较大，即不确定性大，或曰“结合面宽”、“自由运用”。由此可认为“以色列”左右两边是自由的，是一个词
- 邻熵可用来确定语言单位的自由性

单字词的邻字熵

- 约160万字的《人民日报》语料，单字词3978个，左邻字熵平均3.27，右邻字熵平均3.02。（作为对照，非词单字582个，左邻字熵平均0.37，右邻字熵平均0.80）

成词概率	邻字熵		成词概率	邻字熵
把	0.93	15.07	也	0.95	14.93
到	0.49	15.05	以	0.40	14.92
有	0.41	15.05	又	1.00	14.83
向	0.60	15.04	于	0.31	14.73
地	0.33	15.00	后	0.44	14.65

- 成词概率：作为单字词出现的频次 / 该字符的频次
- 邻字熵：左邻字熵+右邻字熵

各种语言的熵

- 英文：4.03
- 法文：3.98
- 西班牙文：4.01
- 俄文：4.35
- 汉字的熵
  - 1970年代末~1984,冯志伟教授手工统计 9.65
  - 通过100多万字的人民日报语料测得9.655
- 语料库规模不同，会有变化。平衡语料库效果更好

点式互信息（pointwise mutual information）

- 点式互信息可用来衡量两事件的相关程度。

$$I(v_1,v_2)=\log\frac{P(v_1,v_2)}{P(v_1)P(v_2)}$$

- 点式互信息为正值（log真数大于1），表明两事件正相关；点式互信息为0（log真数为1），表明两事件无关，点式互信息为负值（log真数小于1），表明两事件互相排斥。
- 上式的约束条件：三个概率均不为0。

点式互信息的计算

- 例：某语料库规模 R＝1606115字次，“昂”出现40次，“扬”出现308次，“昂扬”出现7次，求“昂”和“扬”的互信息。
- 解：I(“昂”；“扬”)

＝log(P(7/R) / P(40/R) / P(308)/R)＝9.83

- 例：语料库同上，“的”出现55202次，“扬的”出现14次，求“扬”和“的”的互信息。
- 解：I(“扬”；“的”)

＝log(P(14/R) / P(308/R) / P(55202)/R)＝0.4

语言模型（Language Model）

◆狭义：度量语言符号串合法性的概率分布。

◆广义：语言信息处理时用来判别或预测的概率公式。

例如，机器翻译的统计模型中，包含一个目标语言模型P(T)和一个翻译模型P(S|T)，其中P(T)就是狭义的语言模型。而下面这个关于机器翻译的公式就是广义的语言模型。

分类（Classification）

- 分类是把样本归入已知类别，是有指导的（分类体系就是一种指导），聚类则是自动建立分类体系并将样本归入这些类别，是无指导的。
- 自然语言处理的根本问题是歧义消解：分词歧义消解、词汇歧义消解、词性歧义消解、句法歧义消解，等等。
- 歧义消解就是分类。例如，已知某兼类词有哪几个词性，要将它在文本中的每个词例一一归类。语言模型的基本作用就是分类，就是消解歧义。

常用的语言模型

- N元模型：一种建立在概率乘法定理之上的语言模型。它假定序列中某个状态的出现只依赖于前面的N-1个状态（马尔科夫假设），据此简化了计算。
- 贝叶斯分类器：一种基于联合概率来处理单点分类问题的语言模型。它忽略上下文的语序和结构，并假定上下文中的各个符号都是独立地起作用的，据此简化了计算。
- 所谓“单点分类”，就是每次只考虑一个符号的分类问题，而不是同时考虑若干个相关符号的分类问题。

常用的语言模型

- 隐马尔科夫模型：一种基于联合概率来处理序列标注问题的语言模型。它包含两个马尔科夫过程，一个是状态转移过程，另一个是由状态到符号的过程，前一个过程是隐藏的。
- 最大熵模型：一种基于条件概率来处理单点分类问题的语言模型。它认为在给定关于训练数据的限制条件下，使模型的熵最大的那个分布，就是所求的分布。
- 条件随机场：一种基于条件概率来处理序列标注问题的语言模型。随机场是指一组随机变量，条件随机场则是以观察序列为条件的随机场。

N元模型

- N元模型，最基本的模型
- N-1阶马尔科夫模型
- 隐马尔科夫模型（HMM）

N元模型

- 句子（词串）是一个状态序列：

  S＝w1w2…wn－1wn

- 由概率的乘法定理可得一个句子的概率为：

  P(S)=P(w1w2…wn－1wn)

  =P(w1)P(w2|w1)P(w3|w1w2)

  … P(wn|w1w2…wn-1)

- 这种计算太复杂，特别是最后一个词的转移概率计算需要考虑前面所有的词。

N元模型

- 设有一个状态序列，N元语法认为，其中某个状态的是否出现只与它前面的 N－1 个状态有关

- 句子（词串）是一个状态序列：

  W＝w1，w2，…，wn－1，wn

- 串的概率为各状态的转移概率之积

“元”（gram）是什么?

- 所谓“状态序列”不一定是词串，还可以是：
- 拼音串，此时“元”是音节
- 汉字串，此时“元”是汉字
- 词性标记串，此时“元”是词性标记
- ……
- N元模型可用于汉字输入法、字音转换、音字转换、自动分词、自动校对、词性标注等。

N元模型的公式表示

- 若N＝1（unigram）则
- P(w1,w2,...,wn)=P(w1)P(w2)...P(wn)

- 若N＝2（bigram）则
- P(w1,w2,...,wn)=P(w1)P(w2|w1)...P(wn|wn-1)

- 若N＝N (历史全需要)，则
- P(w1,w2,…,wn)=P(w1)P(w2|w1)P(w3|w1w2)...P(wn|w1…wn-1)

N元模型的公式表示

- N＝2，则
- P(w1,w2,...,wn)=P(w1)P(w2|w1)...P(wn|wn-1)

$$P(w_1, w_2, \ldots, w_n) = P(w_1)\prod_{i=2}^{n} P(w_i \mid w_{i-1})$$

- N=3，则

$$P(w_1, w_2, \ldots, w_n) = P(w_1)P(w_2 \mid w_1)\prod_{i=3}^{n} P(w_i \mid w_{i-2}, w_{i-1})$$

实际使用的策略

- 由于连乘会导致概率值过小，往往超过了浮点数的表示范围，所以实际使用中，一般采用取对数的方式
- P(w1,w2,...,wn)=P(w1)P(w2|w1)...P(wn|wn-1)
- log P(w1,w2,...,wn)

  =log P(w1)+log P(w2|w1)+…logP(wn|wn-1)

  =logf(w1)-logN+logf(w1w2)-logf(w1)+…+logf(wn-1wn)-logf(wn-1)

实际使用的策略

- 具体的例子（使用人民日报1999年语料抽取数据）
- -25 今天 天气 不错
- Log (P(今天)*P(天气|今天)*P(不错|天气))
- =log f(今天)-logN +logf(今天 天气)-logf(今天) +log f(天气 不错)-logf(天气)
- -53 坦克 洗澡 讨厌
- -76 绿色 的 思想 在 愤怒 地 睡觉
- -65 绿色 的 植物 在 土地 上 成长

N元语言模型的局限

- N元语言模型是局部模型,有些语法现象无法模拟：
- 对One of the life’s best things is a good job，N元模型也许预测are的概率更高

N元模型的用处

◆ 例：字串“我们要使用户满意”的一种切分是

我们 要 使 用户 满意

◆ 若用二元模型，如何计算该词串的概率？

解：P(我们) P(要|我们) P(使|要) P(用户|使) P(满意|用户)

◆ 对计算机来说，该字串还有别的切分，例如：

我们 要 使用 户 满意

◆ 可计算各个词串的概率并加以比较，得出该字串在概率意义上的最佳切分。

N越大越好吗?

- 国标汉字6763个，对字串使用N元模型，则模型参数为6763N个，设N＝2，则有4000多万个参数，即使只要求每对汉字同现一次，也需要一个4000多万字的语料库。但实际上，汉字出现频率是“贫富悬殊”的。
- 4000万字语料中实际出现的双字只有大约100多万种，96%以上的双字从来没有出现过，这就是“数据稀疏”问题 。
- 若N>2，或者“元”不是字而是词甚至词的义项，数据稀疏问题就更加严重。

- 越大越准确（Google的MT）
- 越大计算量越大、数据稀疏越严重

花园里的乌鸦

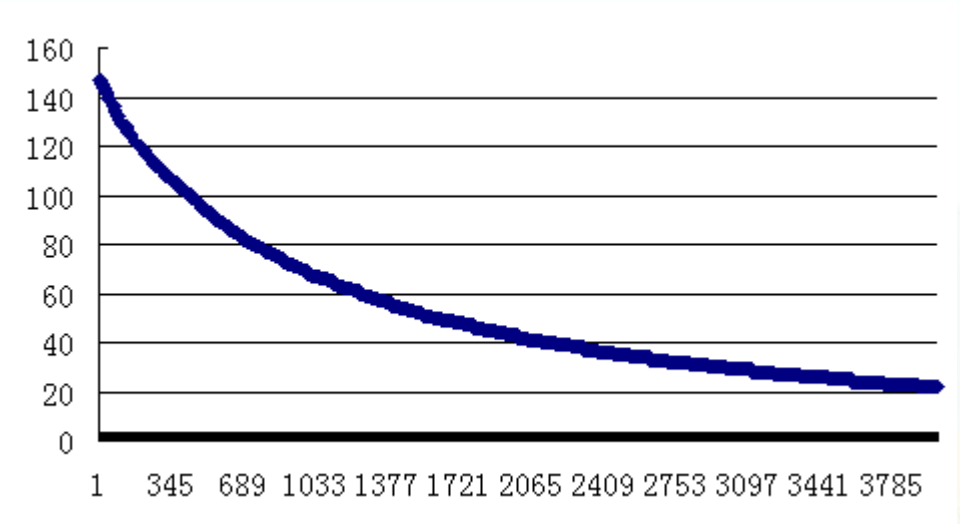
- 假定你看到花园里出现了30种鸟儿，共1000只，其中麻雀212只，知更鸟109只，乌鸦58只，其他鸟儿的数量都较少。请估计：你下一次到这个花园里来，看到一只鸟儿，而这只鸟儿恰好是乌鸦，这件事的概率是多少？
- 你可能会说，这个概率是
- 58/1000=0.058
- 很遗憾，错了！

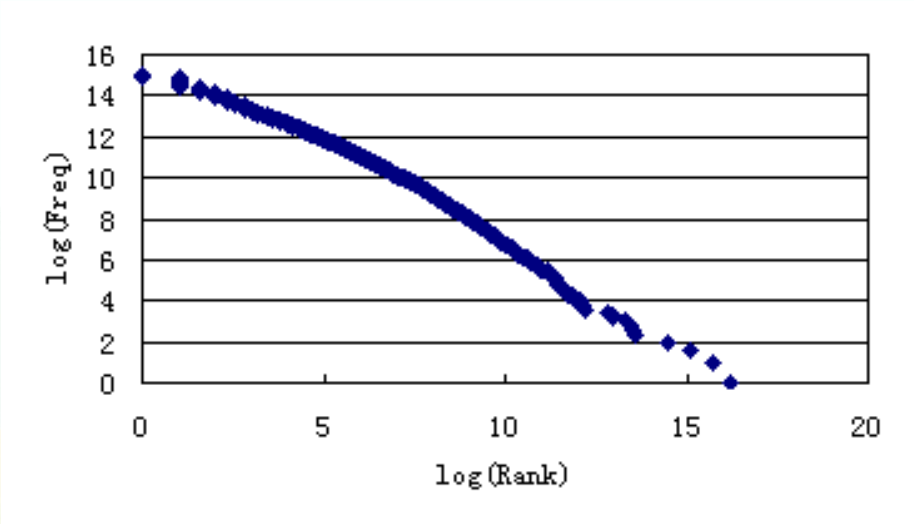
低估和高估

- 下一次飞到花园里来的鸟，可能是第31种、第32种...，也就是说，你这次没见到的鸟，不等于永远不会来。可是，你却把它们的出现概率估计为0.0，这是低估。
- 如果说，没见到的鸟也有一定的出现概率，那么已经见到的鸟的出现概率总和就不应该是1.0，换言之，你对已经见到的这些鸟的出现概率高估了！

齐夫定律(Zipf’s Law)

- 词（型）的频率f与其秩r存在双曲线关系
- **f*r=k**
  - k，为常数
  - log（f）+log（r）=log（k）， y=a-x，趋近于直线
  - 演示计算





定律的意义

- 极少数高频词的出现次数覆盖了整个语料库的绝大部分
- 约一半的词（型）在语料库中仅出现一次
- 这是一种集中效应，也是一种差异分布
- 文字、词、句法规则
- 句法规则在新的语料中会出现，是对Chomsky句法规则有限的一个挑战
- 对于词典编纂和机器学习来说，既有数据垃圾，又存在数据稀疏。
- 对语料库建设的意义：规模问题

参数平滑（parameter smoothing）

- 解决数据稀疏问题的统计方法：
  - 每个事件的出现次数加1，以避免零概率。
  - 设置平伏常数，即以很小的常数代替零概率。
  - 以上两种方法都使得全部事件的概率之和大于1
- Good-Turing方法
  - 将事件的出现次数r调整为r*，平滑后的事件概率为P（E）= r* / N，其中：$r^* = (r+1)n_{r+1}/n_r$
- 插值估计
  - 计算N元条件概率时，也考虑1～N－1元概率，根据经验分配不同的权。

Good-Turning方法举例

	Good-Turing估计				
事件	词条个数	原频率	修正频率	实际词次	估计词次
出现次数为0	50000	0	0.35762	0	17881
出现次数为1	17881	1	0.696829	17881	12460
出现次数为2	6230	2	1.694543	12460	10557
出现次数为3	3519	3	2.687127	10557	9456
出现次数为4	2364	4	3.326988	9456	7865
出现次数为5	1573	5	4.748887	7865	7470
出现次数为6	1245	6	5.66747	7470	7056
出现次数为7	1008	7	7	7056	7056
				72745	79801

语言模型的评估

- 如何评价一个语言模型的好坏
- 有指导方法
  - 有训练语料，有测试语料
  - 正确率（Precision）
  - 召回率（Recall）
  - F值（F-score）
- 无指导方法
  - 无训练语料
  - 交叉熵、困惑度

常规评测

- 一个文本中，有100个重叠式，程序自动识别出120个，其中正确90个，则
- 正确率P = 90 / 120 = 0.75
- 召回率R = 90 / 100 = 0.90

$$F_\beta = (1 + \beta^2) \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{(\beta^2 \cdot \mathrm{precision}) + \mathrm{recall}}$$

- β调节正确率和召回率的关系，如果等同对待，则为1，得到的称F1值
- F1值 = 2*（P*R）/（P+R）

交叉熵（Cross Entropy）

- 设X~p(x)，q(x)是近似于p(x)的一个概率分布，则p(x)与q(x)的交叉熵为：

$$H(p, q) = H(p) + D(p||q) = \sum p(x) (-\log q(x))$$

- 交叉熵反映了两种不确定性：一是真实概率分布本身的不确定性，二是用含有误差的概率分布来近似真实分布所带来的不确定性。

相对熵（Relative Entropy）

- 对于随机变量X，p(x)和q(x)是关于X的两个概率分布，其相对熵定义为：

$$\mathrm{D}(p \,||\, q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}$$

- 也称为Kullback Leibler（KL）距离
- 约定log(0/q)=0, log(p/0)=∞

语言模型的交叉熵

- 设语言的真实分布为p(x)，模型所反映的分布为q(x)，则语言模型的交叉熵就是H(p,q).
- 真实分布p(x)是无法得知的，得另想办法来计算H(p,q)。
- 考虑 H(p, q) = ∑ p(x) (-log q(x))
  - 按模型q(x)求X的每个取值的自信息，按p(x)来求每个自信息的概率加权平均值
  - 如何避开p(x)来求∑ (-log q(x))的平均值。

$$H(T,q) = -\sum_{i=1}^{N} \frac{1}{N} \log_2 q(x_i)$$

语言模型的交叉熵

- 根据信息论的原理，假定语言L的分析或生成是一个稳态的随机过程，是L的一个长度为n的样本，则有

$$H(L,q) = -\lim_{n\to\infty}\frac{1}{n}\log q(x_1^n)$$

- 就是说，当n趋向于无穷大时，语言与模型的交叉熵可以通过简单地计算每个取值的自信息的算术平均值来求得（近似求解）。

语言模型的困惑度

- 困惑度（perplexity）表示一个语言模型处理某个语料库的困难程度：$PP = 2^{H(L,q)}$
- 困惑度越大则难度越大。语言建模的任务就是要设计出困惑度最小的语言模型。因此，困惑度也是评价语言模型性能的一个重要指标。
- 注意：我们是用语料库A训练模型m，用语料库B测试模型m，并计算m对于B的困惑度。

困惑度的语言学意义

- 使用N元模型时，N越大，困惑度越小。
- 同为三元模型，经过参数平滑的，困惑度更小。
- 困惑度越小，说明其性能越好。因此，困惑度是评价语言模型的指标。
- 如果对某语料库所代表的语言一无所知，模型的困惑度达到最大值R，这意味着计算机处理每一个词时都需要对全部R个候选进行计算。