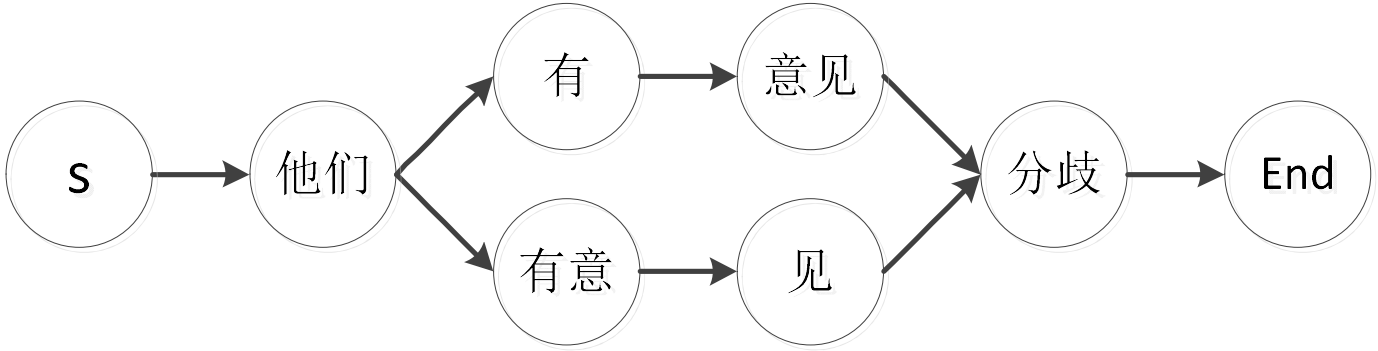

# 最大概率汉语切分&命名实体识别

王小捷
智能科学与技术中心
北京邮电大学

- N-gram语言模型包含了语言知识，如何用？
  - 词预测、句子概率计算
  - 更复杂的任务
    - 用于汉语切分：消除汉语切分的某些歧义
    - 用于信息检索：消除检索中的某些歧义
    - 用于机器翻译：获得更好的翻译结果
    - ……

# 基于N-gram的汉语切分-解决什么问题

- 切分歧义：汉字字串C有多种可能的切分方案，每种方案可以产生一个句子，问题是：那个切分较好？
- 例如：C=他们有意见分歧
  - C1=他们/有意/见/分歧
  - C2=他们/有/意见/分歧



# 基于N-gram的汉语切分-基本想法

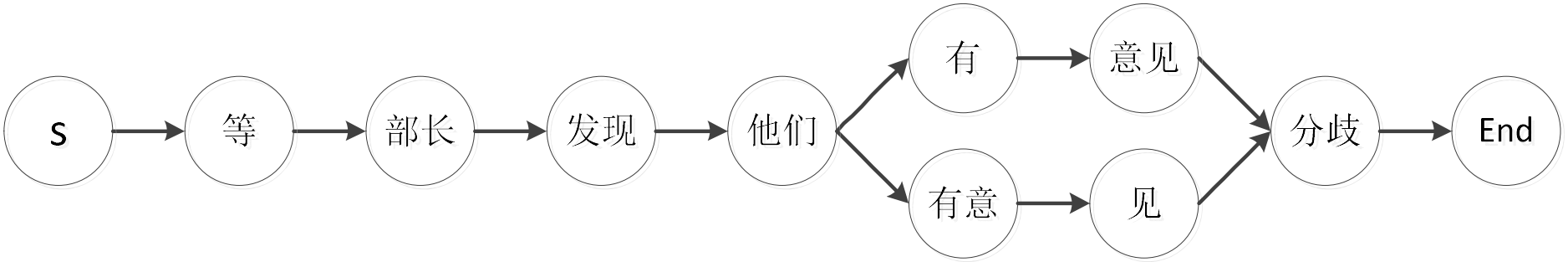
- 1)计算每个可能的句子的概率
- 2)选择概率最大的那个作为切分结果

- 上例中，采用bigram分别计算两个句子概率
  - P(C1)=P(他们/s)P(有意/他们)P(见/有意)P(分歧/见)=0.00018
  - P(C2)= P(他们/s)P(有/他们)P(意见/有)P(分歧/意见)=0.00065
- C2的概率最大，选择C2作为切分结果
- 因此成为最大概率 (MP:Maximum Probability)模型

■问题：复杂一点的句子

■等部长发现他们有意见分歧

■幸运：交集字段为2的基本(>98%)是AB/CD



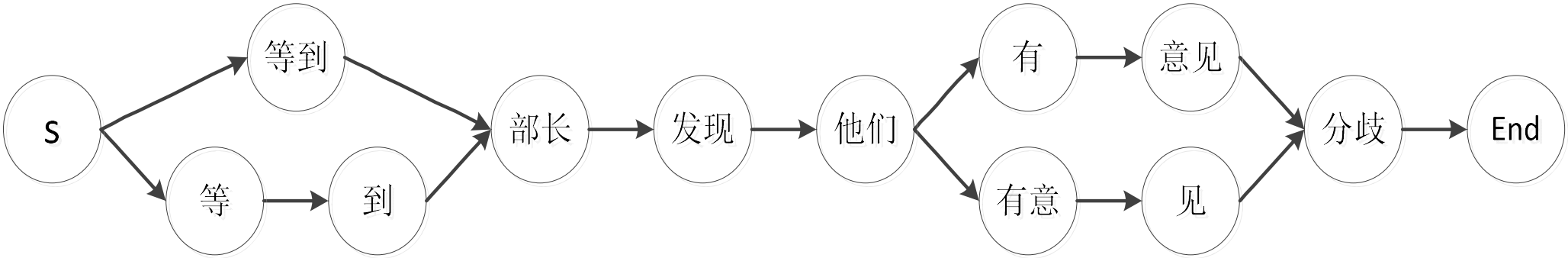

■ P(C1)=P(等/s)P(部长/等)P(发现/部长)P(他们/发现)P(有意/他们)P(见/有意)P(分歧/见)

■ P(C2)= P(等/s)P(部长/等)P(发现/部长)P(他们/发现)P(有/他们)P(意见/有)P(分歧/意见)

■重复计算

■问题：复杂一点的句子

■等到部长发现他们有意见分歧

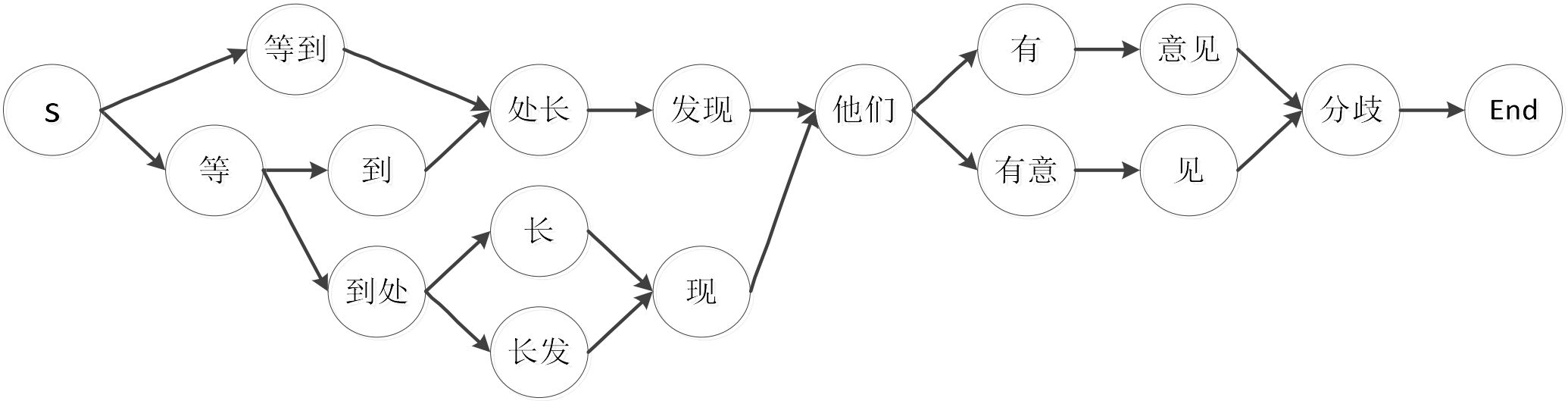



- P(C1)=P(等到/s)P(部长/等到)P(发现/部长)P(他们/发现)P(有/他们)P(意见/有)P(分歧/意见)
- P(C2)=P(等到/s)P(部长/等到)P(发现/部长)P(他们/发现)P(有意/他们)P(见/有意)P(分歧/见)
- P(C3)=P(等/s)P(到/等)P(部长/到)P(发现/部长)P(他们/发现)P(有/他们)P(意见/有)P(分歧/意见)
- P(C4)=P(等/s)P(到/等)P(部长/到)P(发现/部长)P(他们/发现)P(有意/他们)P(见/有意)P(分歧/见)

■更多的重复计算

■问题：再复杂一点

■等*到*处*长*发*现*他们有意见分歧




■ P(C1)=P(等到/s)P(处长/等到)P(发现/处长)P(他们/发现)P(有/他们)P(意见/有)P(分歧/意见)P(End/分歧)=…

■ …

■ P(C8)=

■更多重复的计算

- 保持中间某些中间结果避免重复计算：
  - 中间状态定义
  - 构建基于中间状态的递推运算
- 反向寻优

# 一些术语(以bigram为例)

- 左邻词(Left adjacent word, LAW)
  - $w_{i-1}$ 是$w_i$ 的 LAW
  - 例子：在 “有意见分歧”中
    - Case1(无切分歧义时)：“意见”的LAW :
      - 有/意见/分歧
      - “意见”的LAW 就是一个 “有”
    - Case2(有切分歧义时)： LAW of “分歧”，:
      - 有意/见/分歧：“分歧”的LAW 是 “见”
      - 有/意见/分歧：“分歧”的LAW 是 “意见”
      - “见” 和 “意见” 都是 “分歧”的LAW

# 一些术语(以bigram为例)

- 累积概率(Accumulative Probability, AP)
  - $P_a(w_i)=P_a(w_{i-1})*P(w_i/w_{i-1})$
    - $w_{i-1}$ 是 $w_i$ 的LAW
  - 例子：有意见分歧
    - Case1(unambiguous)： $P_a$(意见)
      - $P_a$(意见)= $P_a$(有)*P(意见/有)，一个累积概率
    - Case2(ambiguous)： $P_a$ (分歧)
      - 有意/见/分歧: $P_a$(分歧from见)=$P_a$ (见)*P(分歧/见)
      - 有/意见/分歧: $P_a$(分歧from意见)=$P_a$ (意见)*P(分歧/意见)
      - 有两个LAW，每一个对应一个累积概率值
- 最大累积概率(Maximum Accumulative Probability, MAP)
  - 只有一个LAW的就只有一个累积概率，MAP就是它
  - 有多个LAW的有多个累积概率，最大的那个就是MAP

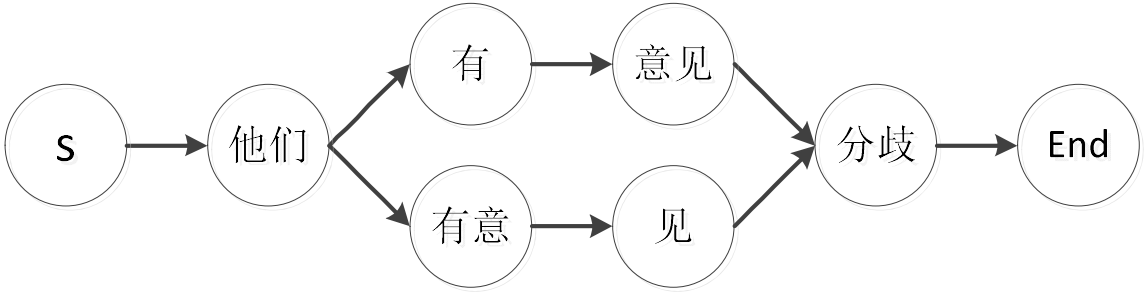
# 一些术语(以bigram为例)

- 最佳左邻词(Best LAW，BLAW)
  - 在 $w_i$ 的几个LAW中，带来最大累积概率(MAP)的LAW
    - 例子：
    - 有意/见/分歧: $P_a$(分歧from见)=$P_a$ (见)*P(分歧/见)
    - 有/意见/分歧: $P_a$(分歧from意见)=$P_a$ (意见)*P(分歧/意见)
  - 如果 $P_a$(分歧from意见)>$P_a$(分歧from见)
  - 那么“意见”为“分歧”的最佳左邻词

# 算法(动态规划)

- 状态：每个词的状态包括两部分
  - (其最大累积概率(MAP), 其最佳左邻词(BLAW))
- 状态计算：
  - 计算每个LAW为该词带来的AP，$P_a(w_i)=P_a(w_{i-1})*P(w_i/w_{i-1})$
  - 比较后保留最大累积概率(MAP)
    - 即：状态= $max_{w_{i-1}} P_a(w_{i-1})*P(w_i/w_{i-1})$
  - 同时可得到该词的最佳(BLAW)
    - w*= $argmax_{w_{i-1}} P_a(w_{i-1})*P(w_i/w_{i-1})$

# 算法(动态规划)

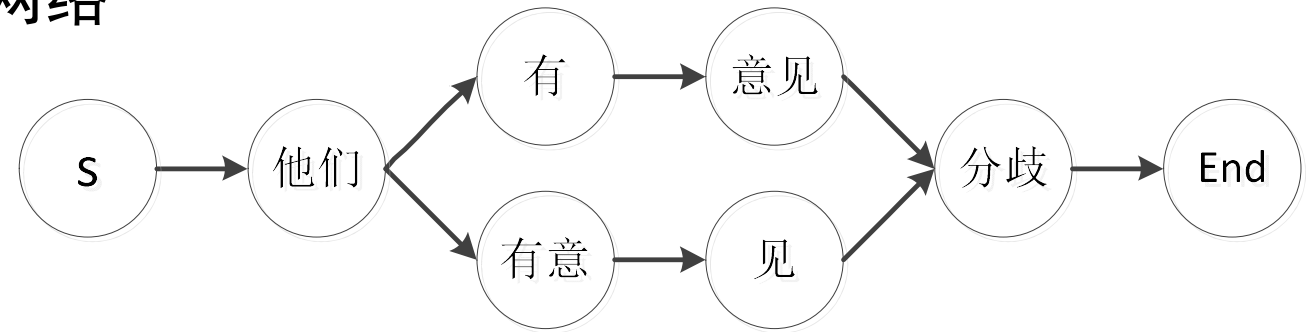



- 初始化:
  - 对输入串, 获取所有的切分候选词，构建成切分词网络。开始词s的状态：(MAP=1，BLAW=ϕ)。
- 前向:
  - 从左到右计算每个候选词的状态(最大累积概率)，并记录每个词的最佳LAW
- 后向:
  - 从右到左，输出每个词的最佳LAW，即得到切分序列

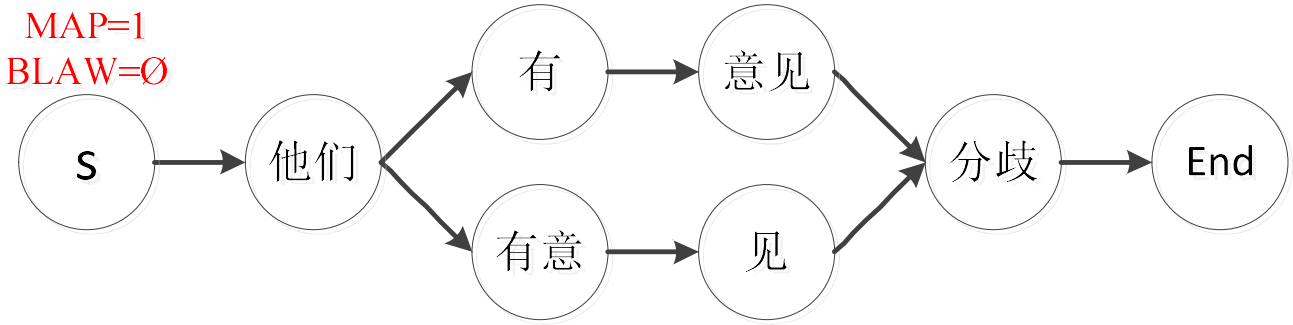
# 算法示例：他们有意见分歧

■初始化：

■1、获得词网络



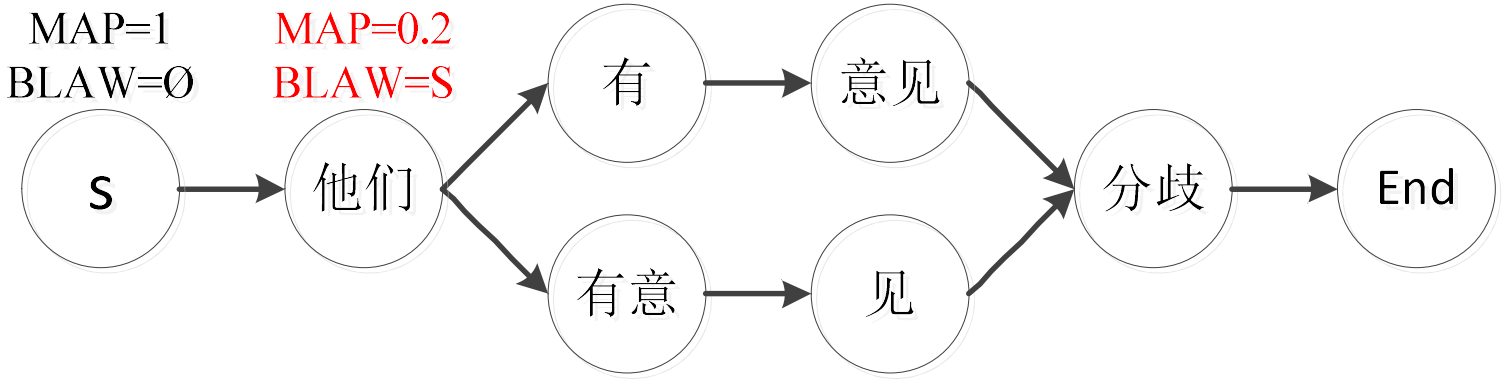

■2、第一个词s的状态



# 算法示例：他们有意见分歧

■前向计算每个节点的状态

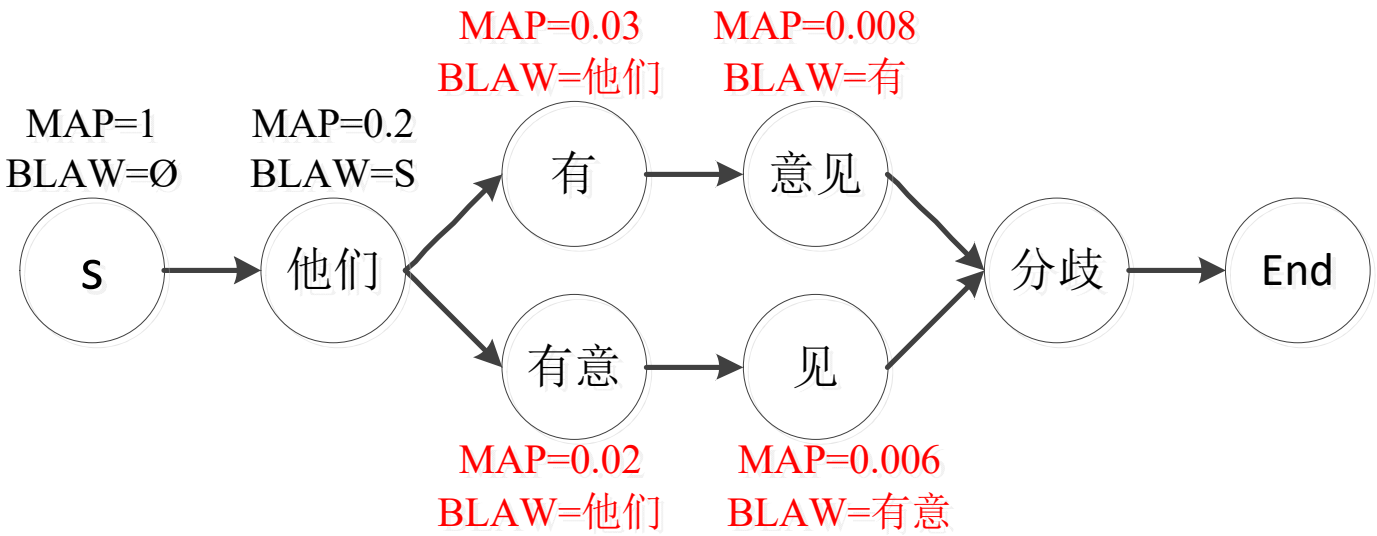
■节点：他们




- Pa(他们)=Pa(S)*P(他们/S)=0.2 (假设)
- 因为当前词只有S一个LAW，无需比较
- 所以它的 MAP=0.2，BLAW=S

# 算法示例：他们有意见分歧

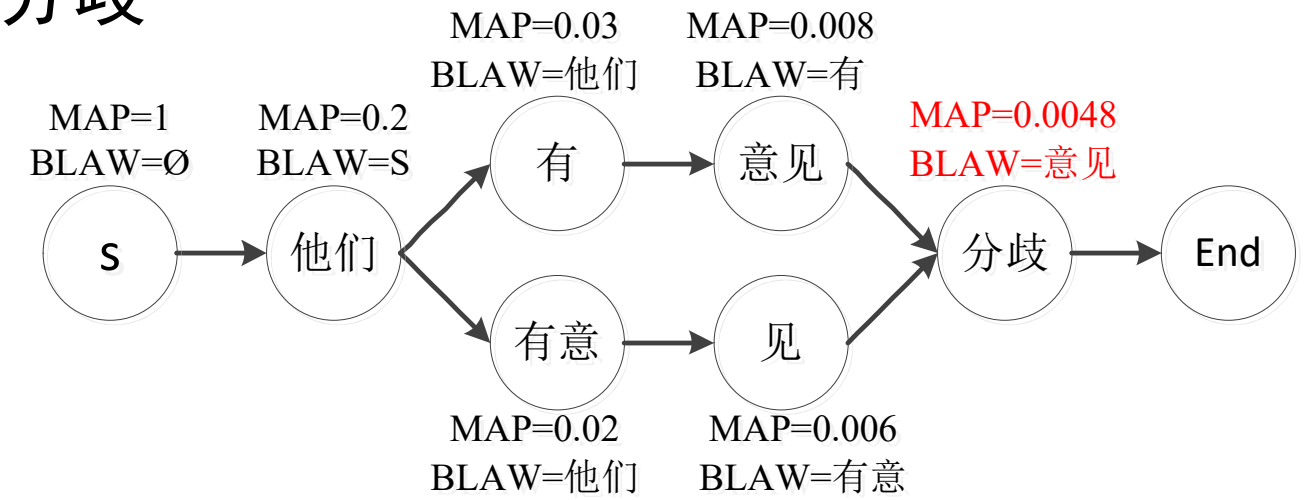
- 前向计算每个节点的状态
- 类似其他节点：有、意见，有意、见




- 每个词的左邻词都只有一个，类似”他们”获得各个词的状态。

# 算法示例：他们有意见分歧

- ■前向计算每个节点的状态
- ■节点：分歧




- 分歧有两个LAW，分别计算其AP：

  Pa(分歧from意见)=Pa(意见)*P(分歧/意见)=0.0048 (假设)

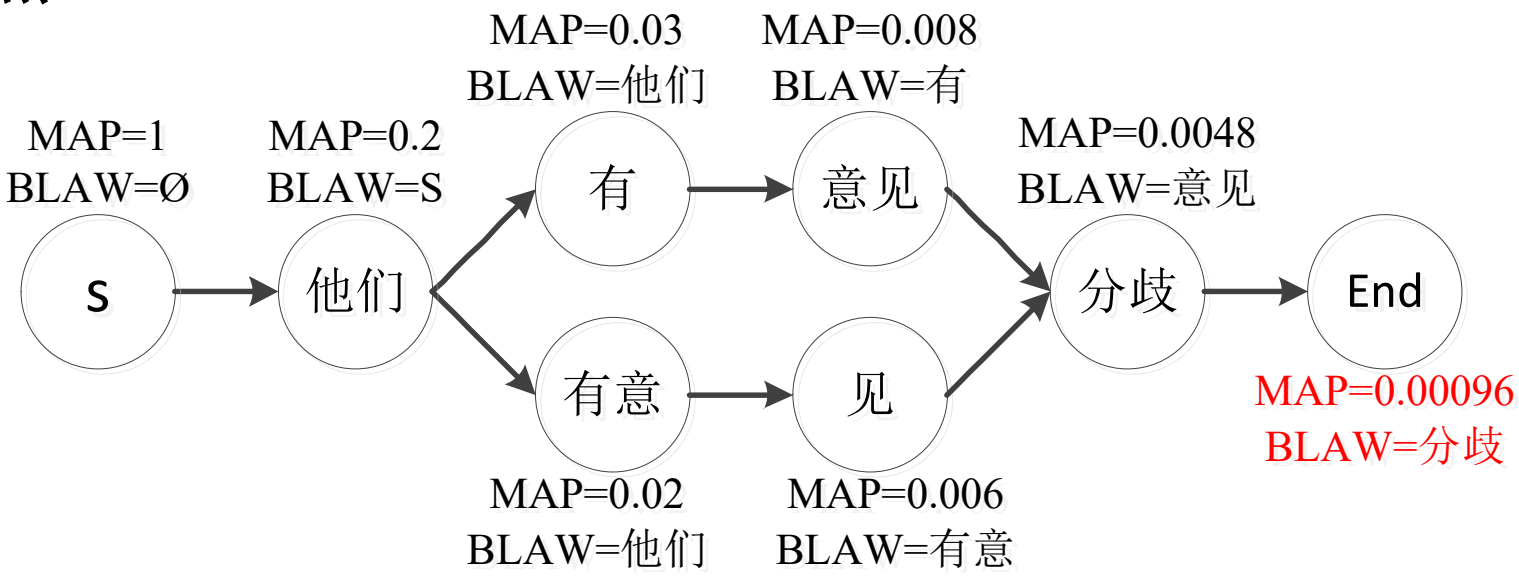
  Pa(分歧from见)=Pa(见)*P(分歧/见)=0.0008 (假设)
- 比较得到：

  MAP=0.0048，BLAW=意见

# 算法示例：他们有意见分歧

■前向计算每个节点的状态

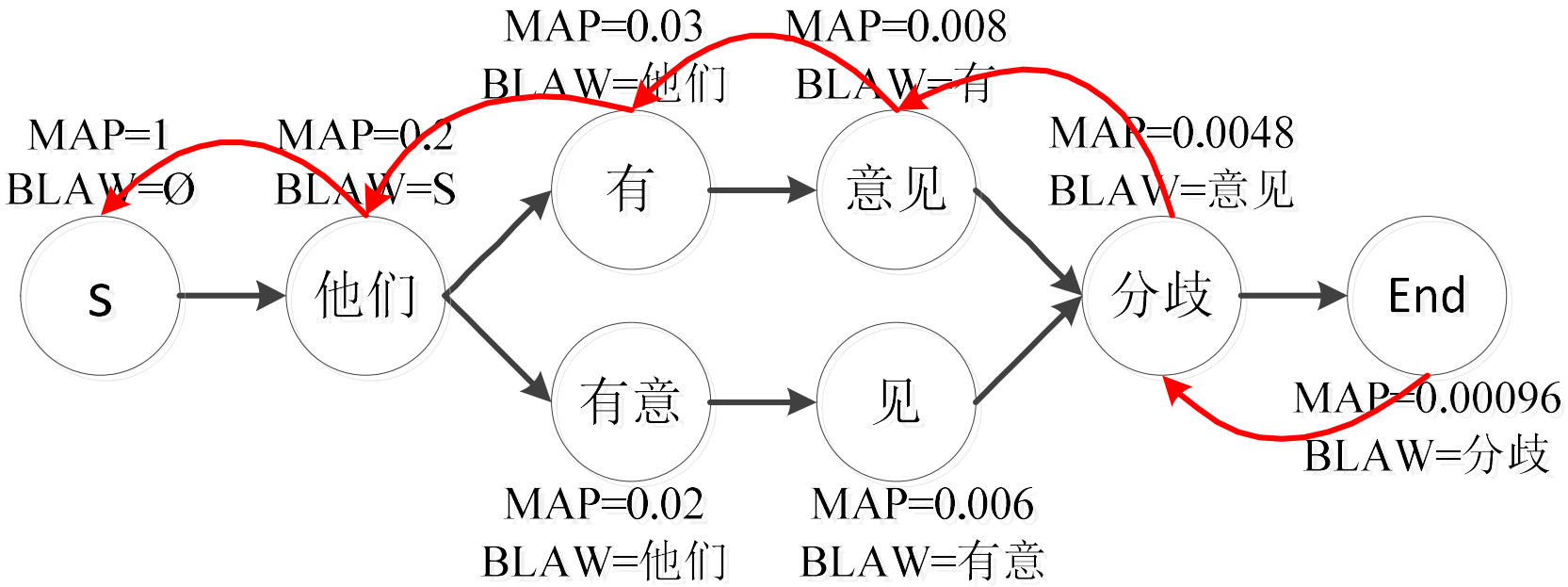
■节点：End




- 只有一个LAW，计算其AP，即为MAP：
- Pa(End from分歧)=Pa(分歧)*P(End/分歧)=0.00096（假设）
- 因此：MAP=0.00096，BLAW=分歧

# 算法示例：他们有意见分歧

■反向通过BLAW获取切分结果



- MP算法处理能力
  - 歧义：能依据语言模型处理切分歧义
    - 交集：有/意见分歧
    - 组合：破获一/起盗窃案件
  - OOV：
    - 李子坚走到桌子前面
      - 语言模型中无数据：包含李子坚的概率均为零

- MP算法问题(影响MP性能的因素)
  - 1、歧义处理能力依赖语言模型
    - 下页例子

# 以unigram语言模型为例

- 交集歧义(Overlap ambiguity)
  - 这事的确定不下来
    - W1= 这/ 事/ 的确/ 定/ 不/ 下来/
    - W2= 这/ 事/ 的/ 确定/ 不/ 下来/
  - 当 Pa(定)<Pa(确定)
- 组合歧义(Combination ambiguity)
  - 做完作业才能看电视
    - W1=做/ 完/ 作业/ 才能/ 看/ 电视/
    - W2= 做/ 完/ 作业/ 才/ 能/ 看/ 电视/
  - 当 Pa(能)<Pa(才能)
- N-gram结果直接影响

- MP算法问题(影响MP性能的因素)
  - 1、歧义处理能力依赖语言模型
  - 2、构建所有可能切分的网络：哪里有歧义，MP本身不能找出歧义，发现歧义需要其他方法帮助
    - 有意见分歧时我们举手表决

- 交集歧义的发现
  - 现象：
    - 交集字段小于2的歧义占总交集歧义的95.41%
    - 交集字段小于3的歧义占总交集歧义的98%
      - 基于500万新闻语料的统计结果[1]
  - 交集起义的发现：链长为奇数(1,3)的交集歧义
    - FMM+ BMM切分结果是不一致的，可以因此侦测到
  - 链长为偶数(2)时
    - 双向最大匹配切分的结果相同，即双向匹配得到的结果都是正确的，不能发现。
    - 但是：98%链长为2的交集歧义字段的正确切分形式是AB/CD [2]。


- [1]周强. 汉语语料库的短语自动划分和标注研究 [学位论文] 北京：北京大学，2002
- [2]闫引堂,周晓强. 交集型歧义字段切分方法研究 情报学报, Vol. 19(6), 2000.12

- 组合歧义的发现
  - [1]在6千万语料：461词(含非双字词)
  - [2]基于1998年人民日报得到双字词组合歧义358个，其中分、合样例均大于5的有66个
    - 不大：不大爱说话；不大于2
    - 正当：正当他进来时；他是正当年
    - 总会：总会下设3个分会；他总会说起这事
    - …


- [1]侯敏，汉语自动分词中的上下文相关歧义字段（CSAS）研究 [A] 孙茂松 陈群秀《自然语言理解与大规模内容计算》[C] 北京：清华大学出版社2005.7
- [2]秦颖，王小捷，张素香，汉语分词中组合歧义字段的研究,中文信息学报，2007，21(1)，.

- 完整的基于MP算法的切分
  - 构建切分网络 (这部分需要外部方法帮助达成)
  - MP算法
    - 前向
      - 计算累计概率，比较获得每个节点的最大累积概率、并记录带来最大累积概率的最佳左邻词
    - 反向
      - 查找每个词的最佳左邻词，获得切分序列


- 基于MP进行汉语切分时利用的知识
  - N-gram相邻词间的同现关系
    - 例如：有意见的话可以提：2-gram,3-gram…
      - 有\意见\的\话\可以\提
      - 有意\见\的\话\可以\提
- 更多的知识还没有在N-gram模型中体现(问题)
  - 例如：远距离相依信息，
    - 有意见的话你们随时都可以通过各种渠道向各级组织提
      - 后有的动词”提”对于将前面切为”意见”提供了重要信息
      - 怎么能用上这个信息？
- 需要更强大的模型去挖掘和利用更多的知识


- 看切分问题的角度分析：
  - 有意见分歧 => 有/意见/分歧
  - $_0$有$_1$意$_2$见$_3$分$_4$歧$_5$：每个位置上是分还是不分的决策
  - 每个位置上进行二分类
    - 例如：有意见的话你们随时都可以通过各种渠道向各级组织提
    - 有0|1意见的话你们随时都可以通过各种渠道向各级组织提
    - 有意0|1见的话你们随时都可以通过各种渠道向各级组织提
- 将切分问题转化为一系列的二分类问题
- 优势：
  - 已有各种有监督分类器：Naïve Bayes、KNN、MaxEnt、DNN…
  - 很多分类器可以使用更多的上下文特征：上例“提”可以作为特征


- 基于有监督分类算法的切分
  - 以基于NB为例：

■ 一、监督二分类问题的定义

■ 已知$(\mathrm{x}_j, y_j), j = 1,2,\dots,N$， $\mathrm{x}_j = (x_{j1}, x_{j2}, \dots x_{jm})$， $x_{ji}$为特征空间的某个特征，$y_j = \{0,1\}$，对于任意新的$\mathrm{x} = (x_1, x_2, \dots x_m)$，$x_i$是相同特征空间的某个特征，求其$y$

■ 二、朴素贝叶斯分类器

■ $y = \underset{y_j}{\operatorname{argmax}}\, p(y_j|\mathrm{x})$

■ $= \operatorname{argmax}_{y_j} p(\mathrm{x}|y_j)p(y_j)/p(\mathrm{x}) = \operatorname{argmax}_{y_j} p(\mathrm{x}|y_j)p(y_j)$

■ $= \operatorname{argmax}_{y_j} p(x_1, x_2, \dots x_m|y_j)p(y_j)$

■ $= \operatorname{argmax}_{y_j} \prod_i p(x_i|y_j)\, p(y_j)$

■ 三、参数估计，如何估计$p(x_i|y_j)$和$p(y_j)$，基于监督数据的极大似然估计

■ $p(x_i|y_j) = \#(x_i, y_j)/\#(y_j)$, $\quad p(y_j) = \#(y_j)/\sum_j y_j$

- 基于NB分类的切分：
  - 训练样本：$(x_j, y_j)$，$x_j = (x_{j1}, \ldots x_{jm})$， $x_{ji}$为词表W=$\{w_1,\ldots w_{|V|}\}$中的词，$y_j = \{0,1\}$
    - ,…,…,一起,…,…,　　y=1
    - ,…,…,一起,…,…,　　y=1
    - …
    - ,…,…,一|起,…,…,　　y=0
    - ,…,…,一|起,…,…,　　y=0
    - …
  - 参数估计
    - $p(w_i|y=1) = \#(w_i, y=1)/\#(y=1)$,
    - $p(w_i|y=0) = \#(w_i, y=0)/\#(y=0)$,
    - $p(y=1) = \#(y=1)/\sum_j y_j$ , $p(y=0) = \#(y=0)/\sum_j y_j$
  - 测试：
    - 新的句子，例“又/一(1|0)起/事故/发生/了”
    - $y = \text{argmax}_{y_j} \prod_i p(x_i|y_j)\, p(y_j)$
    - $y=1$: $\prod_i p(x_i|y_j)\, p(y_j) = p(\text{又}|y=1)p(\text{事故}|y=1)p(\text{发生}|y=1)p(\text{了}|y=1)p(y=1)$
    - $y=0$: $\prod_i p(x_i|y_j)\, p(y_j) = p(\text{又}|y=0)p(\text{事故}|y=0)p(\text{发生}|y=0)p(\text{了}|y=0)p(y=0)$
    - 选择值最大的决策

- 前面NB算法里分类的依据只用了一种特征：
  - 单词(unigram)特征
    - 单词“事故”是否在某种样本中出现
- 更多的可能有用的特征
  - 单词的位置特征
    - 单词“事故”在某种样本中出现时离目标词的距离
  - 两个单词的组合(bigram)
    - 单词“事故”和单词“发生”是否在某种样本中同时出现
  - …
- 越细粒度的特征区分能力就越强
- NB不能利用更多的细粒度特征，更多能利用细粒度特征的分类器
  - 极大熵分类器(MaxEnt)：例如MaxEnt中的特征函数==>
  - 支撑向量机(SVM)
  - …

■例如：MaxEnt中的特征函数

$$p(y \mid x) = \frac{1}{Z_\lambda(x)} \exp(\sum_i \lambda_i f_i(x, y))$$

■可以直接基于语料来设计细粒度特征

■去/一(1)起/事故现场➔ $f_1(x, y) = \begin{cases} 1 & if(w_{-1} = 去，\ w_{+1} = 事故) \\ 0 & else \end{cases}$

■/一(0)起/去事故现场➔ $f_2(x, y) = \begin{cases} 0 & if(w_{+1} = 去，\ w_{+2} = 事故) \\ 1 & else \end{cases}$

- 人工设计特征，陷入特征工程(设计并研究各种特征的作用)
- 近年来基于深度学习的表示学习能力，采用深度学习模型自动学习表示(特征)
  - 向量词表示：比符号的词表示能提供更多词间相似性信息
  - n-gram表示：多个词、句子的表示
  - 多层次抽象表示：词组、句法？
- 学习到的表示(特征)可以作为经典分类器的输入
- 更多的直接用深层神经网络作为分类器，基于分类任务联合训练分类器和表示学习，获得更好的性能。

- 基于分类视角切分的问题
  - $_0$有$_1$意$_2$见$_3$分$_4$歧$_5$
  - 1\2\3\4每个位置都要进行二分类决策，但是，每个分类器都会利用上下文特征，按什么次序决策？先决策会影响后面的决策(上下文变化了)！
  - 联合决策➔基于MP的方法是一种联合决策
  - 但MP需要先有词网络、且只能用LM的知识
  - 既要联合决策，又能自足，并进而利用丰富知识
  - ➔把切分看成是一类序列标注问题

- 处理切分问题角度二：序列标注
  - 标注：给一个对象打标签，标签来自一个预定义标签集L
    - 例如：对象是词，给词打标签，标签集={P，N}
    - 给一个词打标签，可以视为是把标签集作为类标的分类问题
  - 序列标注：给一个有关联的对象序列中的每个对象打标签，标注集L
    - 例如：词序列$w_1, w_2, \ldots w_n$，标签集$L$={Noun，Verb}
    - 则序标问题为：$w_1, w_2, \ldots w_n$➔ $t_1, t_2, \ldots t_n$，其中$t_i \in L, i = 1,2, \ldots n$
    - 多个分类，但是由于对象间有关联，分类时相互影响，因此要联合进行
  - 切分问题转化为序标问题要解决两个问题：对象序列是什么？标签集的设计
  - 1、对象序列是“词序列”还是“字序列”：尚未切分，因此是字序列！
  - 2、标签集设计：为字序列中的每个字打上一个标签后能成为词序列！
    - L={B，I，O}，B：一个词的开始字、I：一个词的非开始字、O：该字单独成词
    - 

| O | B | I | B | I |
|---|---|---|---|---|
| 有 | 意 | 见 | 分 | 歧 |

➔ 有/意见/分歧

- 这样，序列标注算法均可用于做切分，因此切分的进展主要与序标模型的进展关联起来
  - HMM (后面词性标注问题中再具体介绍)
    - HMM只能利用较少的语言知识
  - MEMM
    - 可以用更多特征，综合更多语言知识，但存在模型偏置
  - CRF
    - 更丰富特征，更多语言知识。
    - 2005年，基于字序列标注的CRF模型取得最好切分性能
    - CRF一直到现在仍是序列标注问题的最好模型之一

- 在序列标注框架下其他提升切分性能的发展
  - 不断设计新特征和新标注集以捕获更多的知识
  - 新特征：上下文、成词能力的各类度量等
    - N-gram特征：C-1,C0,C+1,C-1C0,C0C1,C-1C1,…
    - 字符类型特征：数字、单位、标点、字母、中文数字…
    - 结构特征：AAB、AABB、ABAB、…
    - 位置特征：标签更细的词的第几个字…
    - 字的成词能力特征：字C前后不同字的情况…
    - 偏旁部首特征：
  - 新的标注集：BIO、BIEO、$BB_2B_3EO$…
    - E、$B_2$ 、 $B_3$等可以更精细地定位字，发现更细致的知识
    - 例如：某个字更常作为E、某个字更常作为词的第二个字等等这些知识对于切分都有用。
    - 但是，标注集规模增大，增加了标注的难度(例如：从3选1到5选1)

- 基于字序列标记的CRF模型在2005年后就逐渐成为最常用模型，后面的一个研究焦点就集中在特征上了。
- 深度学习带来了深刻的影响，和分类任务一样,从特征工程转变为由深层神经网络自动提取表示。
- 直接基于DNN模型的汉语切分技术的性能与之前的CRF模型相比有所提高，但与DNN在图像、语音信息处理上取得的重大进展相比，还是比较小的。

| 模型 /NLPCC2015数据集 | P | R | F |
|---|---|---|---|
| FDNLP(CRF) | 94.1 | 93.9 | 94.0 |
| GRNN(一种DNN模型) | 94.7 | 94.8 | 94.8 |

- 更多的还是将学到的表示交给CRF模型来进行序标(目前也有某些研究表明无需CRF，只需要简单的softmax分类器，表示层已经学习到足够好的词间关系了)
  - RNN-CRF，LSTM-CRF，BiLSTM-CRF…(词性标注任务时再介绍)
  - BERT-CRF：目前的state-of-the-art模型…

| | PKU | MSR | AS | CITYU | CTB6 | SXU | UD | CNC | WTB | ZX |
|---|---|---|---|---|---|---|---|---|---|---|
| Zhou et al. (2017) | 96.0 | 97.8 | - | - | 96.2 | - | - | - | - | - |
| Yang et al. (2017) | 96.3 | 97.5 | 95.7 | 96.9 | 96.2 | - | - | - | - | - |
| Chen et al. (2017) | 94.3 | 96.0 | 94.6 | 95.6 | 96.2 | 96.0 | - | - | - | - |
| Xu and Sun (2017) | 96.1 | 96.3 | - | - | 95.8 | - | - | - | - | - |
| Yang et al. (2018) | 95.9 | 97.7 | - | - | 96.3 | - | - | - | - | - |
| Ma et al. (2018) | 96.1 | 97.4 | 96.2 | 97.2 | 96.7 | - | 96.9 | - | - | - |
| Gong et al. (2018) | 96.2 | 97.8 | 95.2 | 96.2 | 97.3 | 97.2 | - | - | - | - |
| He (2019) | 96.0 | 97.2 | 95.4 | 96.1 | 96.7 | 96.4 | 94.4 | 97.0 | 90.4 | 95.7 |
| Ours (3 layer) | **96.6** | **97.9** | **96.6** | **97.6** | **97.6** | **97.3** | **97.3** | **97.2** | **93.1** | **97.0** |
| Ours (3 layer+FP16) | 96.5 | **97.9** | 96.4 | 97.5 | 97.5 | **97.3** | **97.3** | 97.1 | 92.7 | **97.0** |

- 更多的还是将学到的表示交给CRF模型来进行序标(目前也有某些研究表明无需CRF，只需要简单的softmax分类器，表示层已经学习到足够好的词间关系了)
  - RNN-CRF，LSTM-CRF，BiLSTM-CRF…
  - BERT-CRF：目前的state-of-the-art模型…

| | PKU | MSR |
|---|---|---|
| Zhou et al. (2017) | 96.0 | 97.8 |
| Yang et al. (2017) | 96.3 | 97.5 |
| Chen et al. (2017) | 94.3 | 96.0 |
| Xu and Sun (2017) | 96.1 | 96.3 |
| Yang et al. (2018) | 95.9 | 97.7 |
| Ma et al. (2018) | 96.1 | 97.4 |
| Gong et al. (2018) | 96.2 | 97.8 |
| He (2019) | 96.0 | 97.2 |
| Ours (3 layer) | **96.6** | **97.9** |
| Ours (3 layer+FP16) | 96.5 | **97.9** |

| PKU数据 | P | R | F | $R_{in}$ | $R_{oov}$ |
|---|---|---|---|---|---|
| 2003 | 0.956 | 0.963 | 0.959 | 0.975 | 0.799 |
| 2005 | 0.969 | 0.968 | 0.969 | 0.976 | 0.838 |

| MSR数据 | P | R | F | $R_{in}$ | $R_{oov}$ |
|---|---|---|---|---|---|
| 2005 | 0.965 | 0.98 | 0.972 | 0.99 | 0.59 |
| 2006 | 0.978 | 0.98 | 0.979 | 0.985 | 0.839 |

- 更多的还是将学到的表示交给CRF模型来进行序标(目前也有某些研究表明无需CRF，只需要简单的softmax分类器，表示层已经学习到足够好的词间关系了)
  - RNN-CRF，LSTM-CRF，BiLSTM-CRF…
  - BERT-CRF：目前的state-of-the-art模型…

| | PKU | MSR |
|---|---|---|
| Zhou et al. (2017) | 96.0 | 97.8 |
| Yang et al. (2017) | 96.3 | 97.5 |
| Chen et al. (2017) | 94.3 | 96.0 |
| Xu and Sun (2017) | 96.1 | 96.3 |
| Yang et al. (2018) | 95.9 | 97.7 |
| Ma et al. (2018) | 96.1 | 97.4 |
| Gong et al. (2018) | 96.2 | 97.8 |
| He (2019) | 96.0 | 97.2 |
| Ours (3 layer) | **96.6** | **97.9** |
| Ours (3 layer+FP16) | 96.5 | **97.9** |

| PKU数据 | P | R | F | $R_{in}$ | $R_{oov}$ |
|---|---|---|---|---|---|
| 2003 | 0.956 | 0.963 | 0.959 | 0.975 | 0.799 |
| 2005 | 0.969 | 0.968 | 0.969 | 0.976 | 0.838 |

| MSR数据 | P | R | F | $R_{in}$ | $R_{oov}$ |
|---|---|---|---|---|---|
| 2005 | 0.965 | 0.98 | 0.972 | 0.99 | 0.59 |
| 2006 | 0.978 | 0.98 | 0.979 | 0.985 | 0.839 |

- 在POS任务时再回到序列标注模型

- 切分性能的发展
  - SIGHAN WS 评测推动汉语切分技术的重要力量
  - 2003,2005,2006,2007,2010,2012,2014
  - NLPCC 2015,2016
  - 2007语料(主要是各类新闻数据)：

| | Training data | | Test data | | OOV |
|---|---|---|---|---|---|
| | Token | Type | Token | Type | |
| CITYU | 1092687 | 43639 | 235631 | 23303 | 19382 |
| CKIP | 721549 | 48114 | 90678 | 14662 | 6718 |
| CTB | 642246 | 42159 | 80700 | 12188 | 4480 |
| NCC | 913466 | 58592 | 152354 | 21352 | 7218 |
| SXU | 528238 | 32484 | 113527 | 12428 | 5815 |

# ■2007 results

| 语料来源 | 评测种类 | 最好F | 最好$F_{IV}$ | 最好$F_{OOV}$ |
|---|---|---|---|---|
| CITYU | Close | 0.9510 | 0.9667 | 0.7698 |
| | Open | 0.9697 | 0.9785 | 0.8750 |
| CKIP | Close | 0.9470 | 0.9623 | 0.7524 |
| | Open | 0.9563 | 0.9692 | 0.7925 |
| CTB | Close | 0.9589 | 0.9697 | 0.7745 |
| | Open | 0.9920 | 0.9936 | 0.9654 |
| NCC | Close | 0.9405 | 0.9573 | 0.6080 |
| | Open | 0.9757 | 0.9800 | 0.8880 |
| SXU | Close | 0.9623 | 0.9752 | 0.7292 |
| | Open | 0.9735 | 0.9820 | 0.8109 |

■切分性能的发展

■对于有较大规模训练数据的规范语言，切分性能达到可用水平

■但并不是每个领域都有这样的大规模标注数据(跨领域研究)

| 领域 | P | R | F | $R_{in}$ | $R_{oov}$ |
|---|---|---|---|---|---|
| 文学 | 0.953 | 0.958 | 0.955 | 0.981 | 0.655 |
| 计算机 | 0.929 | 0.948 | 0.929 | 0.986 | 0.735 |
| 医药 | 0.92 | 0.951 | 0.935 | 0.986 | 0.67 |
| 财经 | 0.95 | 0.964 | 0.957 | 0.983 | 0.763 |

■2010年SIGHAN评测中某系统在各领域的成绩，文学领域最佳，其他领域非最佳。没有系统在所有领域最佳。

- 存在大量非规范用法的网络文本、跨领域成为两个基本问题
  - 2012年的SIGHAN汉语切分技术评测的评测数据来自微博。性能最好的系统取得了P=0.946、R=0.9496和F=0.9478的成绩。但是，整句完全切分正确的比例只有44.88%。
  - 2014年的SIGHAN汉语切分技术评测的评测数据采用的是多领域混合数据。性能最好的系统取得了P=0.9681、R=0.9779和F=0.9730的成绩。

■2015 更复杂的微博数据导致较低的性能

■2016 更复杂的微博数据

| Close | Precision | Recall | F_b |
|---|---|---|---|
| byu-2 （1st） | 0.792917 | 0.816246 | 0.804412 |
| byu-1 （3rd） | 0.7834 | 0.813453 | 0.798144 |

| Semi | Precision | Recall | F_b |
|---|---|---|---|
| byu-2 （1st） | 0.816755 | 0.843353 | 0.829841 |
| byu-1 （2nd） | 0.804658 | 0.835274 | 0.81968 |

| Open | Precision | Recall | F_b |
|---|---|---|---|
| byu-1 （1st） | 0.812359 | 0.832221 | 0.82217 |

# 基本结论

- 汉语切分中歧义的问题已大体解决，未登录词(OOV)的问题依然存在
- OOV的主体是命名实体(Named Entity)
  - 人名:中国人名、外国人名(不同国)、网名、…
  - 地名、组织机构名；
  - 电影名、书名、设备名、药名……
- 命名实体识别(NER，Named Entity Recognition)成为重点问题

- NER
  - 早期设计规则
    - 前缀、后缀、规则……
    - 人名：姓氏字典、名字字典、称呼前缀&后缀、人类行为…
      - 刘英说他今天要去报到。
      - 卧龙先生不在
      - ……
    - 很多例外问题：
      - 词典构建的问题：很多姓氏也是常用字：马上回去、牛劲上来了…
      - 称呼单用：劳烦先生过来一趟
      - ……
  - NER的视角？

■基于序列标注的统计NER

■序标设计：

| P_B | P_I | P_I | O | O | O | L_B | L_I | O | Z_B | Z_I | Z_I | Z_I | Z_I | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 刘 | 修 | 行 | 参 | 观 | 了 | 绍 | 兴 | 的 | 鲁 | 迅 | 纪 | 念 | 馆 | 。 |

■序列标注模型：CRF等

■同样，近年来BERT-CRF取得了最好性能

■NE在不同领域有共性，但更多会具有一些各自不同的特点：

| | 政治 | 体育 | 科学 | … |
|---|---|---|---|---|
| NE | 人名、组织机构名、职名… | 人名、队伍名、赛事名… | 人名、理论名、对象名、 | … |

■NER难点：跨领域的NER

# 总结

- 汉语切分
  - 问题：
    - 歧义：OA、CA
    - 未登录词：OOV
  - 问题视角：基于字的序标标注
  - 模型：序列标注模型
  - 知识：N-gram、更复杂的特征、神经编码表示…

■3：视频中提到可以使用神经网络为序列标注模型自动构建特征，如何实现的呢？【问题频次：1】

■答：以采用BiLSTM(双向长短时记忆)为例，网络结构如下图所示，其中红色框圈出来的部分，是BiLSTM编码得到的每个字位置上的输出表示，这些就是作为神经网络自动构建的特征，这些表示传给上层的CRF模型进行序列标注。