第十章 机器翻译
(2/3)

2、统计机器翻译

10.2.1 统计翻译基本原理

❑ 基本思想

1947年 Weaver 提出“解读密码(解码)”的思想。

1990年 IBM 的 Peter F. Brown 等人在《计算语言学》(Computational Linguistics) 杂志发表的论文“统计机器翻译方法”[Brown，1990]和1993年他们发表在该杂志的“统计机器翻译的数学：参数估计”[Brown，1993]两篇文章奠定了统计机器翻译的基础。

10.2.1 统计翻译基本原理

❑ 噪声信道模型

一种语言T (信道意义上的输入，翻译意义上的目标语言) 由于经过一个噪声信道而发生变形，从而在信道的另一端呈现为另一种语言 S (信道意义上的输出，翻译意义上的源语言)。翻译问题实际上就是如何根据观察到的S，恢复最为可能的T 问题。这种观点认为，任何一种语言的任何一个句子都有可能是另外一种语言中的某个句子的译文，只是可能有大有小[Brown *et. al*, 1990]。

10.2.1 统计翻译基本原理

源语言句子： $S = s_1^m \equiv s_1 s_2 \cdots s_m$

目标语言句子： $T = t_1^l \equiv t_1 t_2 \cdots t_l$

$$P(T \mid S) = \frac{P(T)P(S \mid T)}{P(S)}$$

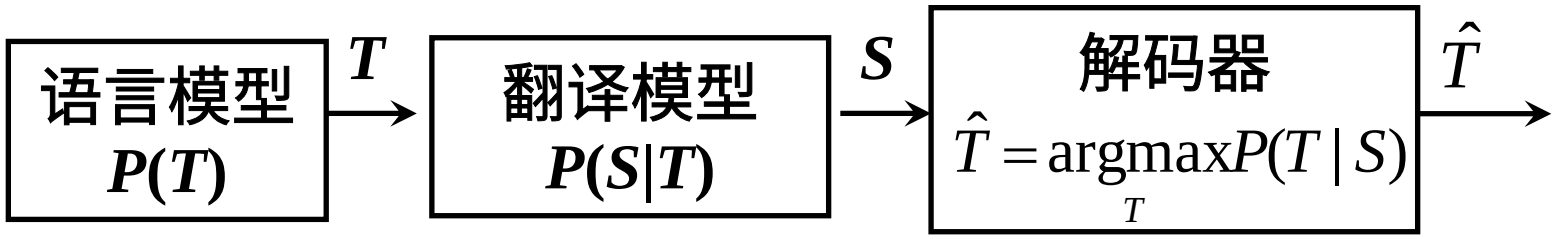
$$\hat{T} = \arg\max_{T} P(T)P(S \mid T)$$

语言模型
Language model, LM

翻译模型
Translation model, TM

10.2.1 统计翻译基本原理

实现搜索过程的模块称为解码器(decoder)。



统计机器翻译中的噪声信道模型

统计翻译中的三个关键问题：

(1) 估计语言模型概率$P(T)$； (2) 估计翻译概率$P(S \mid T)$；

(3) 有效快速地搜索T 使得$P(T) \times P(S \mid T)$最大。

10.2.1 统计翻译基本原理

- 估计语言模型概率 $P(T)$

给定句子：$t_1^l = t_1 t_2 \cdots t_l$

概率：$$P(t_1^l) = P(t_1)P(t_2 \mid t_1)\cdots P(t_l \mid t_1 t_2 \cdots t_{l-1})$$

n-gram 问题，不再赘述。

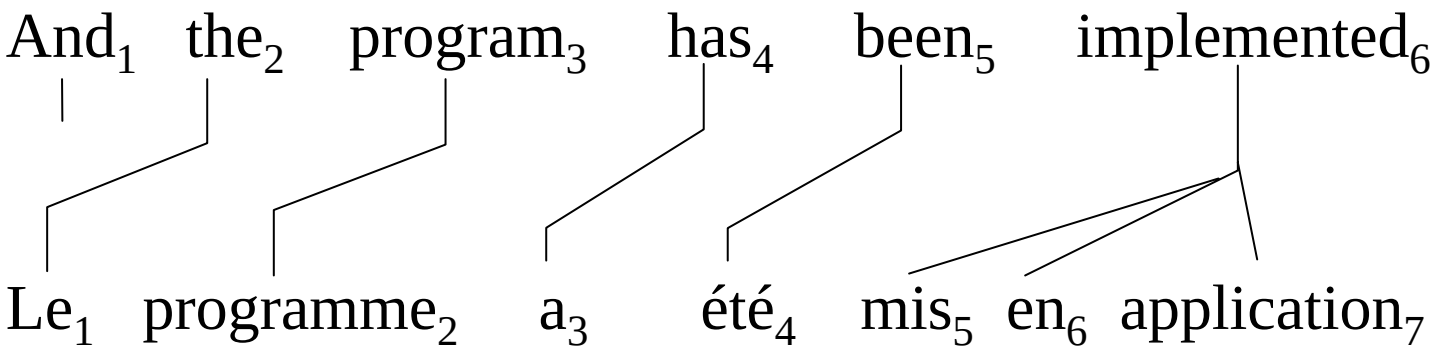
10.2.1 统计翻译基本原理

- 翻译概率$P(S|T)$的计算

关键问题是怎样定义目标语言句子中的词与源语言句子中的词之间的对应关系。

假设英语与法语的翻译对：

***(Le programme a été mis en application | And the* (1) *program* (2) *has* (3) *been*(4) *implemented* (5, 6, 7))**



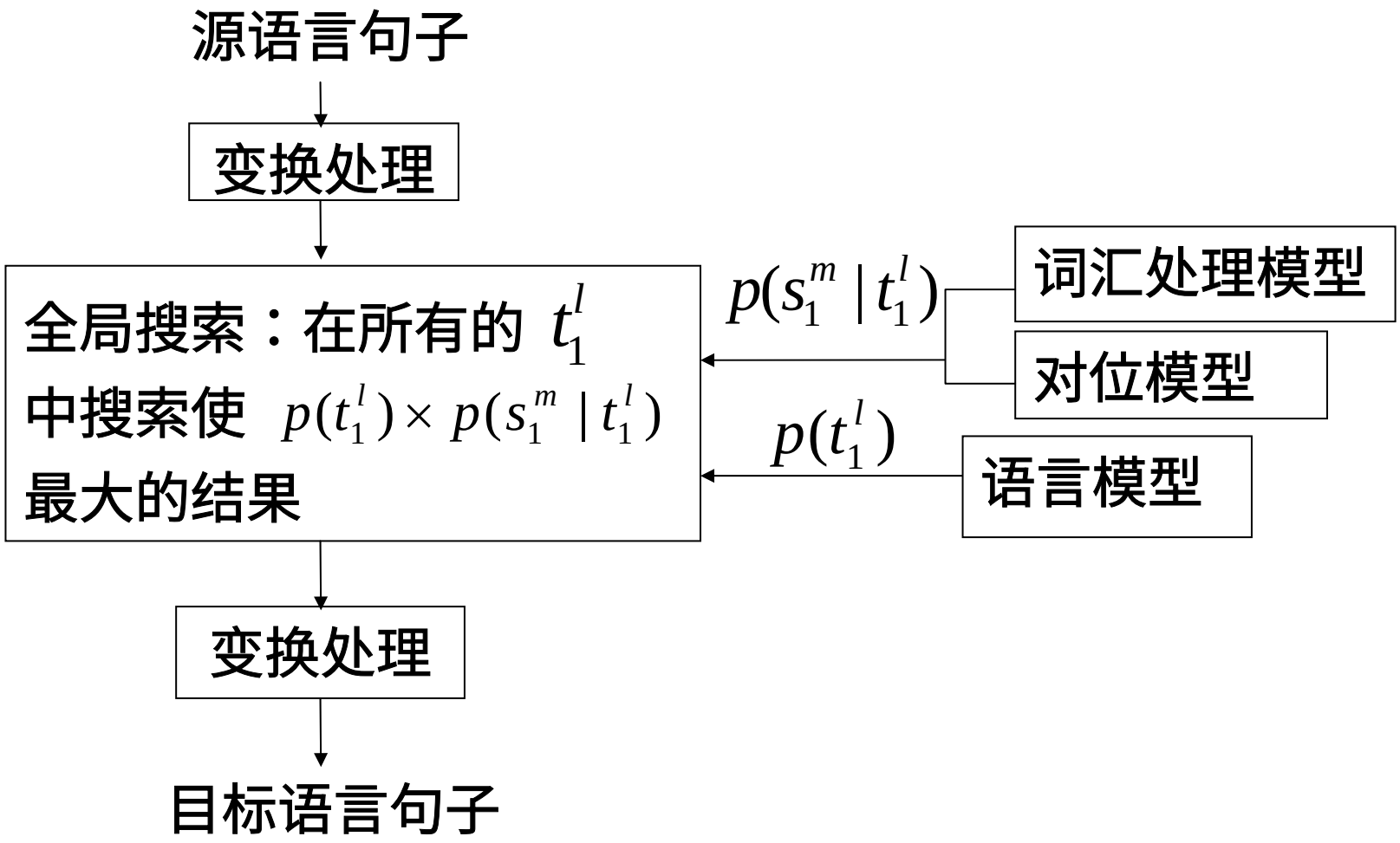
10.2.1 统计翻译基本原理

不妨，我们用$A(S, T)$表示源语言句子S与目标语言句子T之间所有对位关系的集合。在目标语言句子T的长度（单词的个数）为l，源语言句子S的长度为m 的情况下，T和S的单词之间有$l \times m$种不同的对应关系。由于一个对位是由词之间的对应关系决定的，并且不同的对应方式应该是$2^{l \times m}$的子集，因此，$A(S, T)$中共用$2^{l \times m}$ 种对位。

10.2.1 统计翻译基本原理

对于一个给定的句对$(S|T)$，我们可以假定所有的单词对(s_j, t_i)之间存在着对应关系。那么，用来刻画这些对应关系的模型叫做<u>对位模型</u>(alignment model)。

10.2.1 统计翻译基本原理



10.2.1 统计翻译基本原理

❑ 基本的数学问题

求解联合概率分布$P(S=S, A=A, T=T)$，其中，S, T 分别表示源语言和目标语言字符串随机变量，A 为S 与T 之间的对位关系的随机变量。S, A, T 分别表示随机变量 S，A，T 的一个具体取值。

约定用l, m 分别表示目标语言句子的长度和源语言句子的长度，现在我们约定L 和M 分别表示长度l 和m 的随机变量。在不引起混淆的情况下，我们一般用$P(S, A, T)$ 替代 $P(S=S, A=A, T=T)$。

10.2.1 统计翻译基本原理

翻译句对$(S|T)$ 的似然率可以通过条件概率$P(S, A|T)$ 获得：

$$P(S \mid T) = \sum_{A} P(S, A \mid T)$$

按照前面的约定，源语言句子 $S = s_1^m \equiv s_1 s_2 \cdots s_m$ 有m个单词，目标语言句子 $T = t_1^l \equiv t_1 t_2 \cdots t_l$ 有l个单词，对位序列表示成：

$$A = a_1^m = a_1 a_2 \cdots a_m$$

10.2.1 统计翻译基本原理

其中，a_j $(j=1..m)$的取值范围为0到l 之间的整数，如果源语言中的第j 个词与目标语言中的第i个词对齐，那么，$a_j = i$，如果没有词与它对齐，则$a_j=0$。

不失一般性，

$$P(S,A|T)=P(m|T)\prod_{j=1}^{m}P(a_j|a_1^{j-1},s_1^{j-1},m,T)P(s_j|a_1^{j},s_1^{j-1},m,T) \quad (1)$$

实际上，$P(S, A|T)$ 可以写成多种形式的条件概率的乘积，(1)式只是其中的一种。

10.2.2 IBM-1 翻译模型

在上面的(1)式中，由于等号右边有太多的参数，因此，我们不能保证这些参数之间总是互相独立的。因此，在遵循如下三个假设的情况下，我们得到翻译模型1 (记作IBM-1)：

1) 假定$P(m|T)$ 与目标语言T和源语言的句子长度m无关，那么，$\varepsilon \equiv P(m|T)$是一个比较小的常量；

10.2.2 IBM-1 翻译模型

2) 假定 $P(s_j \mid a_1^j, s_1^{j-1}, m, T)$只依赖于目标语言的长度 l，那么，

$$P(a_j \mid a_1^{j-1}, s_1^{j-1}, m, T) = \frac{1}{l+1}$$

3) 假定 $P(a_j \mid a_1^{j-1}, s_1^{j-1}, m, T)$ 仅依赖于s_j和 t_{a_j}。

参数有两个：1) $\varepsilon \equiv P(m \mid T)$ ；2) 给定 t_{a_j}的情况下单词 s_j的翻译概率(translation probability)：

$$p(s_j \mid t_{a_j}) \equiv P(s_j \mid a_1^j, s_1^{j-1}, m, T)$$

10.2.2 IBM-1 翻译模型

如何估计IBM-1的翻译概率？

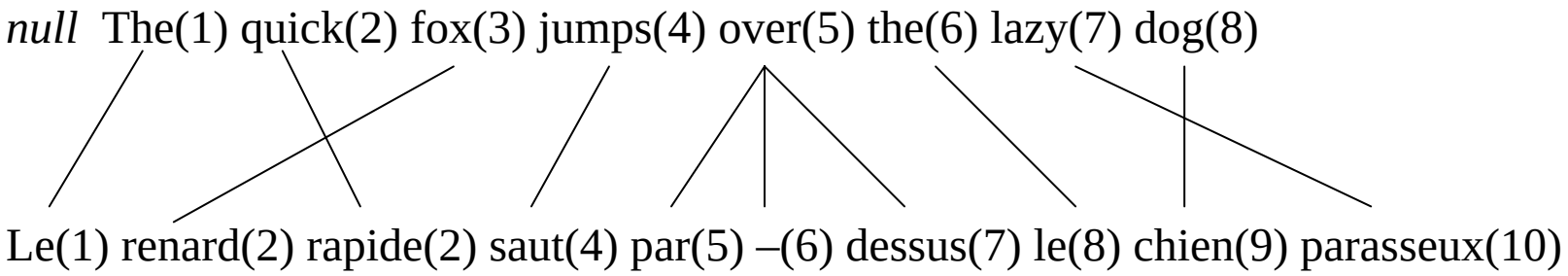
根据(1)式和上面的假设，在给定目标语言句子的情况下，源语言句子和对位关系的联合似然率为：

$$P(S,A|T)=P(m|T)\prod_{j=1}^{m}P(a_j|a_1^{j-1},s_1^{j-1},m,T)P(s_j|a_1^{j},s_1^{j-1},m,T)$$

$$=\frac{\varepsilon}{(l+1)^m}\prod_{j=1}^{m}p(s_j|t_{a_j}) \qquad (2)$$

10.2.2 IBM-1 翻译模型

(2)式的理解：



$$P(S,A|T)=\frac{\varepsilon}{(8+1)^{10}}\times[\underbrace{p(Le|The)\times p(renard|\ fox)\times\cdots\times p(parasseux|\ lazy)}_{\text{共10项}}]$$

10.2.2 IBM-1 翻译模型

由于对位关系由1到m个a_j 的具体值所决定，而每个a_j的取值可以是0到l 之间的任意数，因此，

$$P(S|T)=\frac{\varepsilon}{(l+1)^m}\sum_{a_1=0}^{l}\cdots\sum_{a_m=0}^{l}\prod_{j=1}^{m}p(s_j|t_{a_j}) \qquad (3)$$

我们需要知道所有词对$(s|t)$ 的对应概率p使得翻译概率$P(S|T)$ 最大，并且对于每一个给定的单词t满足以下约束条件：

$$\sum_{s}p(s|t)=1$$

10.2.2 IBM-1 翻译模型

为了求限定条件下概率$P(S \mid T)$ 达到最大值，我们引入拉格朗日乘法因子 λ_t，然后，求下列辅助函数的无限定条件的极大值：

$$h(p,\lambda) \equiv \frac{\varepsilon}{(l+1)^m} \sum_{a_1}^{l} \cdots \sum_{a_m}^{l} \prod_{j=1}^{m} p(s_j \mid t_{a_j}) - \sum_t \lambda_t \left(\sum_s p(s \mid t) - 1 \right)$$

根据求极大值的条件，h函数关于p和 λ 的偏导数应等于零。函数h关于$p(s|t)$的偏导数为：

10.2.2 IBM-1 翻译模型

$$\frac{\partial h}{\partial p(s|t)} = \frac{\varepsilon}{(l+1)^m} \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \sum_{j=1}^{m} \delta(s,s_j)\delta(t,t_{a_j}) p(s|t)^{-1} \prod_{k=1}^{m} p(s_k|t_{a_k}) - \lambda_t$$

其中，δ是 Kronecker 函数，当它的两个参数相同时，$\delta=1$，否则，$\delta=0$。这样，在偏导数等于0情况下，可以求得

$$p(s|t) = \lambda_t^{-1} \frac{\varepsilon}{(l+1)^m} \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \sum_{j=1}^{m} \delta(s,s_j)\delta(t,t_{a_j}) \prod_{k=1}^{m} p(s_k|t_{a_k}) \quad (4)$$

10.2.2 IBM-1 翻译模型

(4) 式给我们提供了一种利用迭代过程求解极大值的思路：给翻译概率一个任意的初始估计值，我们可以计算出等式右边的值，并可以利用这个值作为新的$p(s|t)$ 的估计值。这个重复进行的迭代过程正是期望最大化(Expectation Maximization, EM)算法：

借助于(2)式，我们可以将(4)式写成如下形式：

$$p(s|t) = \lambda_t^{-1} \sum_A P(S, A|T) \sum_{j=1}^{m} \delta(s, s_j)\delta(t, t_{a_j}) \qquad (5)$$

对位A中t 连接到s 的次数

10.2.2 IBM-1 翻译模型

…… ……

忽略详细的数学推导，IBM-1模型表示为：

$$P(S \mid T) = \frac{\varepsilon}{(l+1)^m} \prod_{j=1}^{m} \sum_{i=0}^{l} p(s_j \mid t_i) \qquad (6)$$

10.2.2 IBM-1 翻译模型

根据IBM-1模型，由英语句子e 生成法语句子f 的实现过程：

1) 根据概率分布为法语句子f 选择一个长度m；

2) 对于每一个$j = 1, 2, \ldots, m$，根据均匀分布原则从$0, 1, \ldots, l$ 中选择一个值给a_j；

3) 对于每一个$j = 1, 2, \ldots, m$，根据概率选择一个法语单词f_j。

10.2.3 IBM-2 翻译模型

在IBM-2中，除了假定概率 $P(a_j \mid a_1^{j-1}, s_1^{j-1}, m, T)$ 依赖于位置j、对位关系a_j和源语言句子长度m以及目标语言句子长度l以外，另外两个假设与IBM-1中的假设一样。

引入了对位概率(alignment probabilities)的概念：

$$a(a_j \mid j, m, l) \equiv P(a_j \mid a_1^{j-1}, s_1^{j-1}, m, l)$$

10.2.3 IBM-2 翻译模型

对于每一个三元组(j, m, l)，对位概率满足如下约束条件：

$$\sum_{i=0}^{l} a(i \mid j, m, l) = 1$$

类似于IBM-1的推导，得到IBM-2模型：

$$P(S \mid T) = \varepsilon \prod_{j=1}^{m} \sum_{i=0}^{l} p(s_j \mid t_i) a(i \mid j, m, l) \qquad (7)$$

如果对位概率设为常数，IBM-2模型退化为IBM-1模型，即模型1是模型2的特例。

10.2.3 IBM-2 翻译模型

根据 IBM-2模型，由英语句子***e*** 生成法语句子***f*** 的实现过程：

1) 根据概率分布为法语句子f 选择一个长度m；

2) 对于每一个$j = 1, 2, \ldots, m$，根据概率分布 $a(a_j \mid j, l, m)$ 从$0, 1, \ldots, l$ 中选择一个值给a_j；

3) 对于每一个$j = 1, 2, \ldots, m$，根据概率选择一个法语单词f_j。

10.2.4 IBM-3 翻译模型

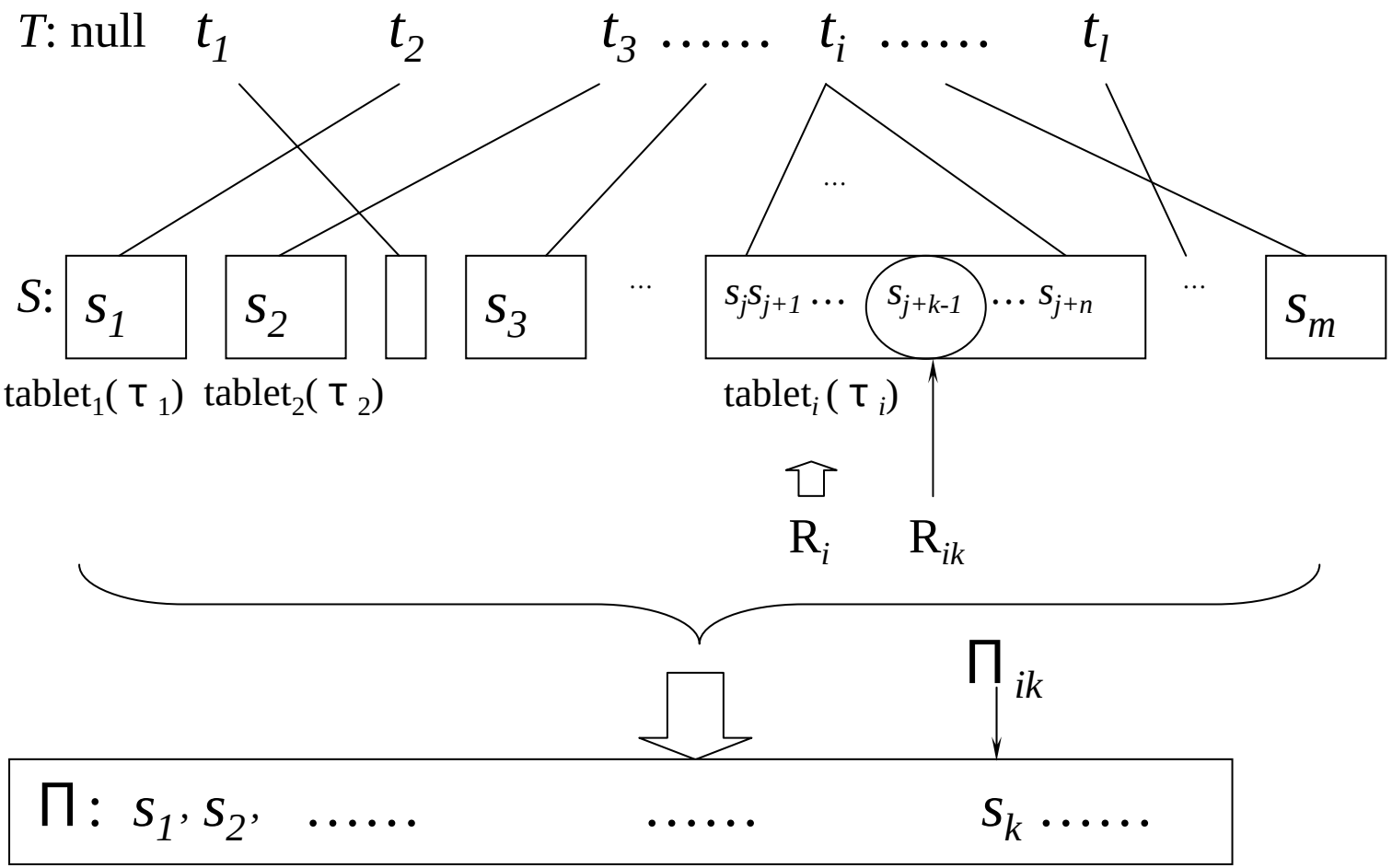
定义：在随机选择对位关系的情况下，与目标语言句子中的单词t对应的源语言句子中的单词数目是一个随机变量，不妨记做Φ_t，我们称该变量为单词t的繁衍能力或产出率(**fertility**)。一个具体的取值记做：ϕ_t

实际上，所谓的繁衍能力就是目标语言单词与源语言单词之间一对多的关系。

10.2.4 IBM-3 翻译模型

定义：假设给定一个目标语言句子T，T中的每一个单词t在源语言句子中可能有若干个词与之对应，源语言句子中所有与t对位的单词列表我们称之为t的一个便笺或写字板(tablet)，这个便笺可能为空。一个目标语言句子T的所有便笺的集合是一个随机变量，我们称之为T的便笺集或标释集(tableau)，记做符号R。T的第i个单词的便笺也是一个随机变量，不妨记做R_i，那么，T的第i个单词的便笺中第k个源语言单词也是一个随机变量，我们记做R_{ik}。

10.2.4 IBM-3 翻译模型



10.2.4 IBM-3 翻译模型

标释集 τ (R的一个具体取值) 和单词排列 π(Π 的一个具体取值，即τ中单词的一种排列方式) 的联合似然率为：

$$P(\tau,\pi\mid T)=\prod_{i=1}^{l}P(\phi_i\mid\phi_1^{i-1},T)P(\phi_0\mid\phi_1^{l},T)\times$$

繁衍概率 (fertility prob.)

$$\prod_{i=0}^{l}\prod_{k=1}^{\phi_i}P(\tau_{ik}\mid\tau_{i1}^{k-1},\tau_0^{i-1},\phi_0^{l},T)\times$$

翻译概率 (tran. prob.)

$$\prod_{i=1}^{l}\prod_{k=1}^{\phi_i}P(\pi_{ik}\mid\pi_{i1}^{k-1},\pi_1^{i-1},\tau_0^{l},\phi_0^{l},T)\times$$

位变概率 (distortion prob.)

$$\prod_{k=1}^{\phi_0}P(\pi_{0k}\mid\pi_{01}^{k-1},\pi_1^{l},\tau_0^{l},\phi_0^{l},T)\qquad(8)$$

10.2.4 IBM-3 翻译模型

假设：

1) 对于1到l中的每一个i，概率$P(\phi_i \mid \phi_1^{i-1}, T)$ 仅依赖于ϕ_i 和 t_i，记作：$n(\phi \mid t_i) \equiv P(\phi \mid \phi_1^{i-1}, T)$

2）对于所有的i，概率 $P(\tau_{ik} \mid \tau_{i1}^{k-1}, \tau_0^{i-1}, \phi_0^l, T)$ 只依赖于τ_{ik}和t_i，记作： $p(s \mid t_i) \equiv P(R_{ik} = s \mid \tau_{i1}^{k-1}, \tau_0^{k-1}, \phi_0^l, T)$

3）对于1到l中的每一个i，$P(\pi_{ik} \mid \pi_{i1}^{k-1}, \pi_1^{i-1}, \tau_0^l, \phi_0^l, T)$ 只依赖于π_{ik}，i, m和l。位置概率记作：

$$d(j \mid i, m, l) \equiv P(\Pi_{ik} = j \mid \pi_{i1}^{k-1}, \pi_1^{i-1}, \tau_0^l, \phi_0^l, T)$$

10.2.4 IBM-3 翻译模型

设想 τ_1^l 中每个写字板中的一组词都存在一个额外词(即这个词在对齐时对空)，假设这个额外词出现的概率为p_1。另外，由于 $\phi_0+\phi_1+\cdots+\phi_l=m$ 因此，

$$P(S|T)=\sum_{a_1=0}^{l}\cdots\sum_{a_m=0}^{l}P(S,A|T)$$

$$=\sum_{a_1=0}^{l}\cdots\sum_{a_m=0}^{l}\binom{m-\phi_0}{\phi_0}(1-p_1)_0^{m-2\phi_0}\,p_1^{\phi_0}\prod_{i=1}^{l}\phi_i!n(\phi_i|t_i)\times\prod_{j=1}^{m}p(s_j|t_{a_j})d(j|a_j,m,l) \quad (9)$$

10.2.4 IBM-3 翻译模型

其中，

$$\sum_s p(s \mid t) = 1$$

$$\sum_j d(j \mid i, m, l) = 1$$

$$\sum_\phi n(\phi \mid t) = 1$$

估计这些参数和 p_1。

10.2.4 IBM-3 翻译模型

根据 IBM-3模型, 一个英语句子e 翻译成法语句子f 的工作过程如下：

1) 对于英语句子中的每一个单词e，选择一个产出率ϕ，其概率为 $n(\phi|e)$；

2) 对于所有单词的产出率求和，得到 *m-prime*；

3) 按照下面的方式构造一个新的英语单词串：删除产出率为0的单词，复制产出率为1的单词，复制两遍产出率为2的单词，依此类推；

4) 在这*m-prime*个单词的每一个后面，决定是否插入一个空单词NULL，插入的概率为p_1，不插入的概率为p_0；

10.2.4 IBM-3 翻译模型

5) 设ϕ_0为插入空单词NULL的个数；

6) 设m为目前总单词的个数：$m\text{-}prime + \phi_0$；

7) 根据概率表 $p(f \mid e)$ 将每一个单词e替换为法语单词f ；

8) 对不是由空单词NULL产生的每一个法语单词，根据概率表$d(j \mid i, l, m)$ 赋予一个位置。这里j是法语单词在法语句子中的位置，i是产生当前这个法语单词的英语单词在其句子中的位置，l是英语句子的长度，m是法语句子的长度；

9) 如果任何一个法语句子的位置被多重登录（含一个以上的单词），则失败返回；

10.2.4 IBM-3 翻译模型

10) 给空单词NULL产生的法语单词在句子中赋予一个位置，这些位置必须是没有被占领的空位置。任何一个赋值都被认为是等概率的，概率值为$1/\phi_0$；

11) 读出法语单词串，其概率为上述每一步概率的乘积，按概率大小输出结果。

10.2.5 IBM-4/5 翻译模型

- **IBM-4:** 考虑标释集的中心词的概率和其它单词的位置概率。
- **IBM-5:** 源语言句子单词间的相对位置。
- **请参见文献:** Peter F. Brown *et al*. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2): 263-311.

10.2.6 基于结构的翻译模型

- 基本思想 [Wang Yeyi, 1998]
  - 粗对齐(rough alignment)：类似IBM-2模型；
  - 细对齐(detailed alignment)：类似IBM-4模型。

两种语言的短语之间利用粗糙模型对位，在训练语料上，通过基于互信息的双语词语聚类和短语归并迭代，得到基于词语聚类的短语规则，利用这些规则进行短语分析，然后对短语内部的单词细对齐。(Structure-based alignment model)

10.2.6 基于结构的翻译模型

方法描述

给定英语句子e，翻译成法语句子f：

(1) 对e 进行短语分析，得到e 的短语序列E；

(2) 根据$p(q|e, E)$ 决定 f 的短语个数q ；

(3) 计算每个法语短语与英语短语对齐的概率;

(4) 对每个法语短语E_i，通过概率分布决定该短语在法语句子中的初始位置；

10.2.6 基于结构的翻译模型

(5) 通过细化的对位模型生成法语短语中的单词，计算每个英语单词的繁衍概率;

(6) 根据英语单词的翻译率生成法语单词写字板;

(7) 根据概率分布对写字板中的法语单词进行排列，决定每个单词在句子中的位置。

实验证明，基于结构的翻译模型由于在不同层次使用不同的模型，因此，达到了降低模型复杂度、扩大分析单元粒度、引入结构信息的目的，并且提高了翻译的正确率。

10.2.7 基于句法的翻译模型

- 基本思想 [Yamada et al., 2001]

基于结构的翻译模型无法处理短语的嵌套结构。Syntax-based SMT 将句法分析树(而不是句子)作为噪声信道模型的输入，信道模型对句法分析树的每个节点进行相应的操作。

10.2.7 基于句法的翻译模型

- 方法描述

1)将句法树扁平化处理(相同中心词的多层结点压缩到一层)；

2)句法树上每个结点的子结点进行随机重新排列(reordering)，每种排列方式都有一个概率；

10.2.7 基于句法的翻译模型

3) 对于句法树任何一个位置随机地插入一个新的目标语言单词(inserting)，每个位置、每个被插入的单词都有不同的概率；

4) 对句法树上每个叶结点的源语言单词翻译成目标语言(translating)，每个不同的译文词都有不同的概率；

5) 根据句子概率(上述概率乘积)输出句子。

11.2.8 基于短语的翻译模型

- 基本思想 [Koehn, 2003] (Phrase-based)

将一个法语句子f翻译成英语句子e：

$$\arg\max_e p(e \mid f) = \arg\max_e p(f \mid e)p(e)$$

考虑到位置的调整:

$$e_{best} = \arg\max_e p(f \mid e)p(e)\omega^{length(e)}$$

ω为长度因子，一般ω>1，倾向于较长的输出。

11.2.8 基于短语的翻译模型

- 模板对位模型 [Ney, 2000; Och, 2002等]
  - 采用基于类的模型，解决数据稀疏问题。运用双语词聚类技术，将两种语言的每一个词都对应到一个类中，共计使用了400个类。
  - 在语言模型上，采用基于类的 5-gram，back-off 数据平滑方法；

11.2.8 基于短语的翻译模型

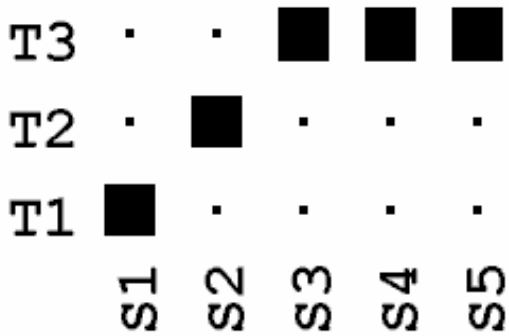
- 在翻译模型上，提出了对齐模板(alignment template)的方法，实现两种层次的对齐：短语层次的对齐和词语层次的对齐，对齐模板采用基于类的对齐矩阵形式表示；
- 对齐模板自动获取，所有可能的模板都保留，但概率不同。如，模板：

  2006年3月15日 关于假冒…==> X 关于假冒…

10.2.8 基于短语的翻译模型

- 用反向的翻译模型替代正向的翻译模型：

$$p(e \mid f)$$



T1: zwei, drei, vier, fünf, ...
T2: Uhr
T3: vormittags, nachmittags, abends, ...

S1: two, three, four, five, ...
S2: o'clock
S3: in
S4: the
S5: morning, evening, afternoon, ...

10.2.9 基于最大熵的翻译方法

- **基本思想** [Och, 2002等] (ME Approach)
  - 反向的翻译模型用噪声信道模型无法解释。
  - 对于一个随机事件，假设已经有了一组样例，我们希望建立一个统计模型来模拟这个随机事件的分布。
  - 选择一组特征，使得统计模型在这一组特征上与样例中的分布完全一致，同时保证这个模型尽可能的“均匀”（使模型的熵值达到最大）。

10.2.9 基于最大熵的翻译方法

- 假设T、S分别是目标语言句子和源语言句子，$h_1(T, S), \ldots, h_M(T, S)$ 分别是T、S上的M个特征，$\lambda_1, \ldots, \lambda_M$ 是这些特征分别对应的M个权值。

对于给定的源语言句子S, 其最佳译文T 可以用以下公式表示：

$$\hat{e} = \arg\max_{T} \{P(T \mid S)\}$$

$$= \arg\max_{T} \{\sum_{m=1}^{M} \lambda_m h_m(T, S)\}$$

10.2.9 基于最大熵的翻译方法

如果我们将两个特征分别取为$\log P(T)$ 和 $\log P(S|T)$，并取$\lambda_1 = \lambda_2 = 1$，那么，这个模型就等价于噪声信道模型，基于噪声信道模型的方法只是基于最大熵方法的一种特例。最大熵方法中最常用的做法就是采用二值特征。

$$\hat{e} = \arg\max_{T} \{\sum_{m=1}^{M} \lambda_m h_m(T, S)\}$$

10.2.10 其它翻译模型

- 基于反向转换文法(inversion translation grammar, ITG)的翻译模型 [Wu, 1997];
- 层次化的基于短语的翻译模型[Chiang, 2005];
- 基于语块的翻译模型(chunk-based) [Watanabe, 2003];
- 基于有限状态转换机的翻译模型[Alshawi, 2000];

…… ……

10.2.11 解码算法

- 基于堆栈的解码算法

(stack-based decoding algorithm, best-first)

1) 用空串作为假设初始化栈，并将各假设的记分初始化为零；

2) 从栈中弹出具有最高记分的元素，记为 current-hypothesis；

3) 如果current-hypothesis 是一个完整的句子，则输出该句子并终止该过程，否则执行下一步；

10.2.11 解码算法

4) 从词汇表中选择一个单词附加到current-hypothesis 末端以扩展该假设，计算并记录新假设的记分，然后将新的假设压入栈内。对于词汇表中的所有单词进行本步操作；

5) 转第（2）步。

10.2.11 解码算法

假设汉语句子g 翻译成英语句子e，那么，翻译模型为 $p(g|e)$，语言模型为$p(e)$。栈内的一个假设为英语句子，不妨表示成如下形式：

$$H = l: e_1 e_2 \ldots e_k.$$

其中，l 为英语句子的长度，$e_1 e_2 \ldots e_k$ 表示英语句子的前k个单词。假设H 的记分f_H 由两部分构成：一部分为句子前段$e_1 e_2 \ldots e_k$ 的打分g_H；另一部分为有待于附加到H上的词串$e_{k+1} e_{k+2} \ldots e_l$ 的启发函数h_H，即 $f_H = g_H + h_H$。

10.2.11 解码算法

- g_H 的计算方法

根据IBM-2翻译模型，有

$$P(g|e)=\varepsilon\sum_{a_1=0}^{l}\cdots\sum_{a_m=0}^{l}\prod_{j=1}^{m}p(g_j|e_{a_j})a(a_j|j,l,m)$$

$$=\varepsilon\prod_{j=1}^{m}\sum_{i=0}^{l}p(g_j|e_i)a(i|j,l,m) \qquad (10)$$

对于假设，每个英文单词e_i对目标语言句子中的单词 g_j的概率贡献为 $\varepsilon p(g_j|e_i)a(i|j,l,m)$

10.2.11 解码算法

对于每个假设 $H = l: e_1 e_2 \ldots e_k$，我们用 $S_H(j)$ 表示假设中的词对目标语言单词g_j 贡献的概率量：

$$S_H(j) = \varepsilon \sum_{i=0}^{k} p(g_j \mid e_i) a(i \mid j, l, m) \qquad (11)$$

为了将分值计算变成加法运算，这里对公式(10)采用了对数运算，这样，翻译模型对句子前段的记分g_H贡献为：

$$\sum_{j=0}^{m} \log S_H(j)$$

10.2.11 解码算法

由于翻译的目的是让概率 $P(e|g)$ 最大化，因此，语言模型的概率也必须考虑，于是，对于N元语法，有

$$g_H = \sum_{j=0}^{m} \log S_H(j) + \sum_{i=0}^{k} \log P(e_i \mid e_{i-N+1} \cdots e_{i-1}) \qquad (12)$$

而上述假设$H = l: e_1 e_2 \ldots e_k$.的评价记分$g_H$可以通过该词串的前身假设 $P = l: e_1 e_2 \ldots e_{k\text{-}1}$ 计算得出，即

$$g_H = g_P + \Delta$$

10.2.11 解码算法

而

$$\Delta = \log P(e_k \mid e_{k-N+1} \cdots e_{k-1}) + \sum_{j=0}^{m} [\log S_H(j) - \log S_P(j)]$$

$$S_H(j) = S_P(j) + \varepsilon p(g_j \mid e_k) a(k \mid j, l, m)$$

因此，

$$g_H = g_P + \log P(e_k \mid e_{k-N+1} \cdots e_{k-1}) + \sum_{j=0}^{m} \log \left[1 + \frac{\varepsilon p(g_j \mid e_k) a(k \mid j, l, m)}{S_P(j)} \right] \quad (13)$$

10.2.11 解码算法

为了避免栈内假设在早期时 $S_P(j)$ 趋于0的情况，可以有两种处理办法，一是在利用EM算法进行参数估计时限制在假定的零位置上空单词的翻译概率要大于某个给定的阈值；另一种办法是给初始的空假设H_0赋一个很小的概率π 以确保不趋于0。王野翊的实验中，$\pi = 10^{-4}$ 时得到了最好的结果。

10.2.11 解码算法

- 启发式函数h_H的计算方法

假设中尚未实现部分的语言模型的分值：

$$h_H^{LM} = -(l-k)PP_{train} + C \quad (14)$$

常数

在训练数据上的困惑度

当 $k << l$ 时，(14) 式主要取决于$-(l-k)PP_{train}$ 的值；k 趋于l 时，主要取决于C 的值。

10.2.11 解码算法

为了估计翻译模型的值，引入变量$v_{il}(j)$ 表示对 i 和 l 之间任何位置上可能的英语单词e 翻译到汉语单词g_j 的概率的最大贡献：

$$v_{il}(j) = \max_{i \le k \le l, e \in L_E} p(g_j \mid e) a(k \mid j, l, m) \qquad (15)$$

其中，L_E 表示英语词汇。

10.2.11 解码算法

当$k < l$时，对于假设$H = l: e_1\ e_2\ \ldots\ e_k$的启发函数为：

$$h_H = \sum_{j=1}^{m} \max\{0, \log(v_{(k+1)l}(j)) - \log S_H(j)\} - (l-k)PP_{train} + C \quad (16)$$

一个新词能够给第j个英语单词带来的概率的最大增量。

当$k=l$时，由于再没有单词可以追加到栈内的假设上去，因此，$h_H=0$。

10.2.11 解码算法

◆ 两点说明

(1)考虑到计算机存储空间的限制，可以设置了一个常数M，当栈内假设数超过M时，搜索算法将对分值最低的假设自行剪枝，在王野翊的实验中，M＝20000。另外，考虑到系统执行时间的限制，设置常数T，一旦解码器的搜索时间超过常量T，算法将推出搜索，宣告解码失败。在他们的实验中，T＝6000，大约相当于两个半小时。

(2)为了避免解码器总是优先选择长句子假设，可以采用多栈搜索[Magerman, 1994] 。

10.2.11 解码算法

- 柱搜索(beam search) [Ney, 92; Tillmann, 03]

  基本思想：集中搜索最可能的假设，使用两种剪枝技术：1) 覆盖位置剪枝；

  2) 集合基数剪枝。

- Pharaoh 解码器 [Koehn, 2003]

  (http://www.isi.edu/licensed-sw/pharaoh/)

10.2.11 解码算法

以西班牙语到英语翻译为例：

Maria	no	daba	una	bofetada	a	la	bruja	verde

Mary not give a slap to the witch green

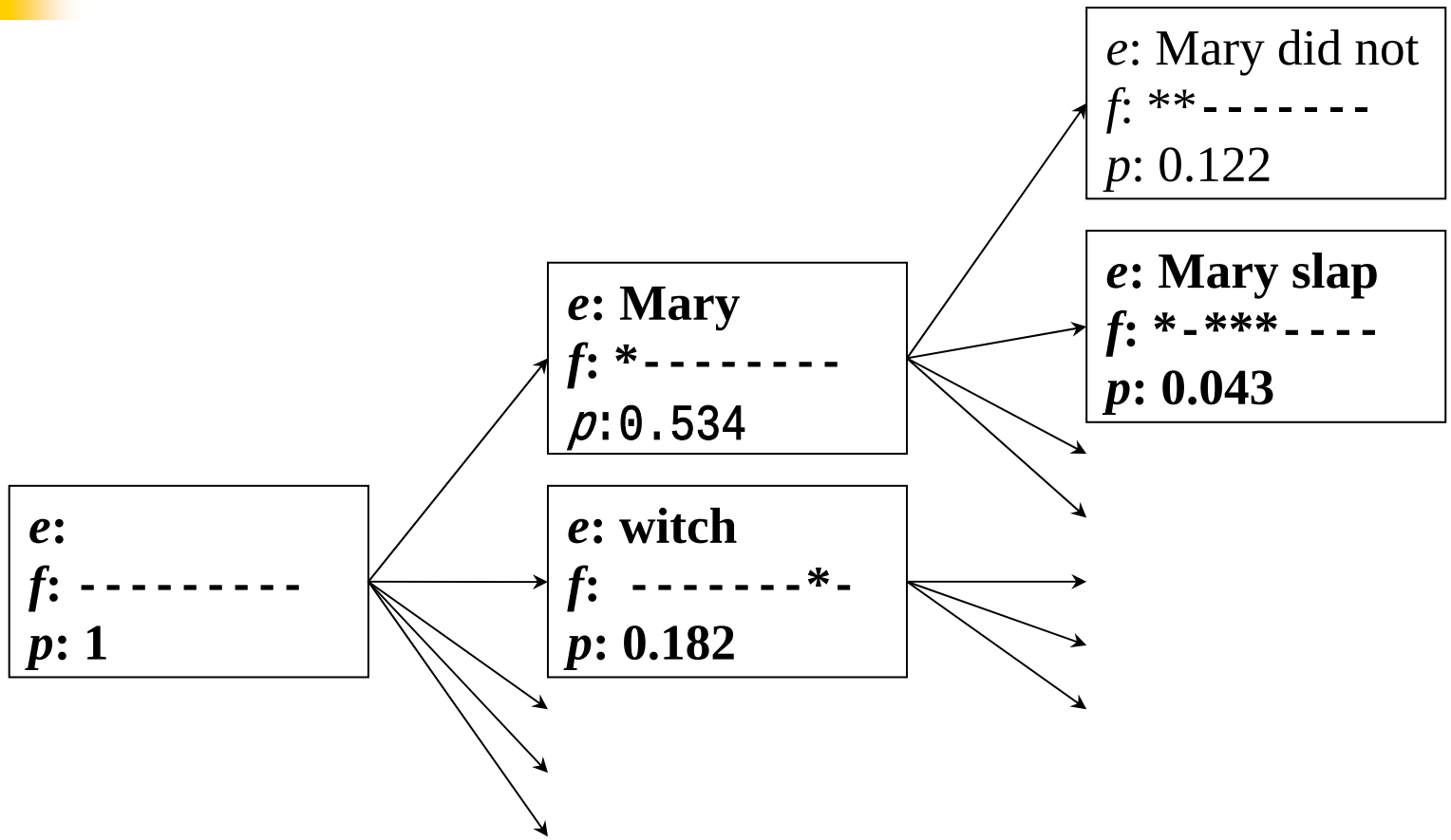
did not a slap by green witch

no slap to the

did not give to

slap the witch

10.2.11 解码算法


e: Mary did not
f: **-------
p: 0.122
e: Mary slap
f: *-***----
p: 0.043
e: Mary
f: *--------
p:0.534
e:
f: ---------
p: 1
e: witch
f: -------*-
p: 0.182

10.2.11 解码算法

(1) 初始化假设栈 hypothesisStack[0 .. nf];

(2) 创建初始假设 hyp_init; 将假设压入栈hypothesisStack[0]

(3) 从i=0到i = nf-1执行如下循环：

对于栈hypothesisStack[i]中的每一个假设hyp执行如下循环{

对于 hyp 可生成的每个新假设new_hyp 执行操作{

nf[new_hyp]= 被new_hyp覆盖的源语言单词个数;

把新假设new_hyp压栈 hypothesisStack[nf[new_hyp]];

对栈hypothesisStack[nf[new_hyp]]进行剪枝;}}

(4) 从栈hypothesisStack[nf]找到最好的假设best_hyp;

(5) 输出产生best_hyp的最佳路径;

10.2.11 解码算法

- 其它搜索方法

(1) Franz J. Och等A*搜索算法(基于IBM-4)[Och, 2001];

(2) Ulrich Germann 等人曾基于IBM-4 构造了贪心爬山(greedy hill-climbing) 解码算法[Germann, 2001; 2003]；

(3) 双向搜索[Watanabe, 2002]，等。

10.2.12 SMT系统实现

(1)语料准备和预处理

工具：EGYPT- TokenizeE.perl.tmpl / 汉语分词

(2)工具准备：

语言模型：SRI LM/ CMU-Cambridge LM

翻译模型工具：GIZA++

解码器：Pharaoh / ReWrite

EM 最小错误率参数训练工具

(http://www.cs.cmu/ashish/mer.html)

10.2.12 SMT系统实现

(3)模型构建与系统集成

模块的结构化、系统的开放性、接口规范性

(4)系统调试

系统运行测试、性能测试、参数调试

10.2.13 译文评估方法

- 常用的评测指标
  - 主观评测：
    1）流畅度：主观评分
    2）忠实度：主观评分
  - 客观评测
    1) 句子错误率：译文与参考答案不完全相同的句子为错误句子。错误句子占全部译文的比率。

10.2.13 译文评估方法

2) 单词错误率：译文与最接近的参考答案之间的编辑距离的比率 (*Multiple Word Error Rate*, 记作mWER)。

3) 与位置无关的单词错误率 (*Position independent mWER*, 记作mPER)

4) 运用基于**unigram** 的**F-**测度计算文本的相似度(GTM)

10.2.13 译文评估方法

- BLEU评价方法 **[Papineni, 2002]**

  －BiLingual Evaluation Understudy, IBM

  基本思想：

  将机器翻译产生的候选译文与人翻译的多个参考译文相比较，越接近，候选译文的正确率越高。

10.2.13 译文评估方法

实现方法：

统计同时出现在系统译文和参考译文中的N元词的个数，最后把匹配到的N元词的数目除以系统译文的单词数目，得到评测结果。

10.2.13 译文评估方法

$$BLEU = BP \times \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

最大语法的阶数，实际取4。

长度过短句子的惩罚因子

w_n=1/N

出现在答案译文中的*n*元词语接续组占候选译文中*n*元词语接续组总数的比例。

$$BP = \begin{cases} 1 & if \quad c > r \\ e^{(1-r/c)} & if \quad c \le r \end{cases}$$

c为候选译文中单词的个数，***r*** 为答案译文中与***c*** 最接近的译文单词个数。

10.2.13 译文评估方法

NIST评估方法

- National Institute of Standards and Technology, USA

基本思想：

在BLEU方法的基础上提出来的，如果一个N元词在参考译文中出现的次数越少，表明它所包含的信息量越大，那么，它对于该N元词就赋予更高的权重。

10.2.13 译文评估方法

- 统计翻译中的译文错误

1) 模型错误：概率最高的译文不是正确的。

2) 搜索错误：概率最高的译文是正确的，但搜索算法找不到。这类错误大约占5%。

10.2.14 统计翻译的相关活动

- John Hopukins University (JHU) 的夏季研讨班

1) 1999年夏天开始

2) 开发了原代码公开的统计翻译根据包－EGYPT

-GIZA：从双语语料中抽取统计模型知识(参数训练);

-Decoder：执行翻译过程；

-Cairo：系统的可视化界面，用于管理所有的参数、查看双语语料对齐的过程和翻译模型的解码过程。

-Whittle：语料预处理工具。

10.2.14 统计翻译的相关活动

- IWSLT: International Workshop on Spoken Language Translation
  - 从2004年开始，每年一次；
  - 旅游领域的口语翻译和新闻领域文本翻译。
- **DARPA 的机器翻译评测活动**
  - **2002年6月, NIST 首次正式的机器翻译评估。**
  - 每年一次。

本讲小结

- 统计机器翻译的基本原理
- IBM 1~5 翻译模型
- 翻译模型的改进
- 解码算法
- SMT 系统实现
- 关于译文评估
- 统计翻译的相关活动

Thanks

谢谢!