

Natural Language Processing
#L11
Machine Translation

袁彩霞

yuancx@bupt.edu.cn

智能科学与技术中心

- 机器翻译
- 噪声信道模型
- IBM model 1
- IBM model 2
- 基于短语的MT及其解码
- 基于循环神经网络的机器翻译

机器翻译：例子

黛玉自在枕上感念宝钗...又听见窗外竹樵蕉叶之上，雨声淅沥，清寒透幕，不觉又滴下泪来。(The Story of the Stone, Cao Xueqin 1792)

Google翻译： Po Chai Daiyu comfortable pillow gratitude ... and I heard on the window bamboo firewood banana leaf, rain pattering, raw cold through the curtain, I feel they drop tears.

霍克斯翻译： As she lay there alone, Dai-yu's thoughts turned to Bao-chai... Then she listened to the insistent rustle of the rain on the bamboos and plantains outside her window. The coldness penetrated the curtains of her bed. Almost without noticing it she had begun to cry.

- 人工智能领域中的最古老问题之一
- AI-hard: 推理及世界知识获取

机器翻译中的歧义

- 正确的翻译必须解决语法及语义歧义
 - "John **plays** the guitar." → "约翰**弹**吉他."
 - "John **plays** soccer." → "约翰**踢**足球."
- 一个早期的英-俄机器翻译系统的例子：
 - "The spirit is willing but the flesh is weak." ⇒ ... (Russian) ⇒ "The liquor is good but the meat is spoiled."
 - "Out of sight, out of mind." ⇒ ... (Russian) ⇒ "Invisible idiot."

机器翻译中的语言因素

- 语言形态不一：

- **黏着型语言**：词内有专门表示语法含义的附加成分，一个附加成分对应于一种语法意义，一种语法意义基本上由一个附加成分表达，词根或词干跟词的附加成分结合不紧密（**芬兰语、日语、蒙古语**）
- **分析型语言**：词基本上没有表示语法含义的附加成分，词的形态变化很少，语法关系靠词序和虚词表示（**汉语、藏语**）
- **曲折型语言**：用词的形态变化表示不同的语法关系，一个形态可以表达几个不同的语法关系，词根或词干跟词的附加成分结合很紧密，往往不能截然分开（**英语、德语**）

机器翻译中的语言因素

- 句法变换不一：
 - **SVO** (e.g. Chinese): 我是留学生
 - **SOV** (e.g. Japanese): 私は留学生です
 - **VSO** (e.g. Arabic): أنا طالب

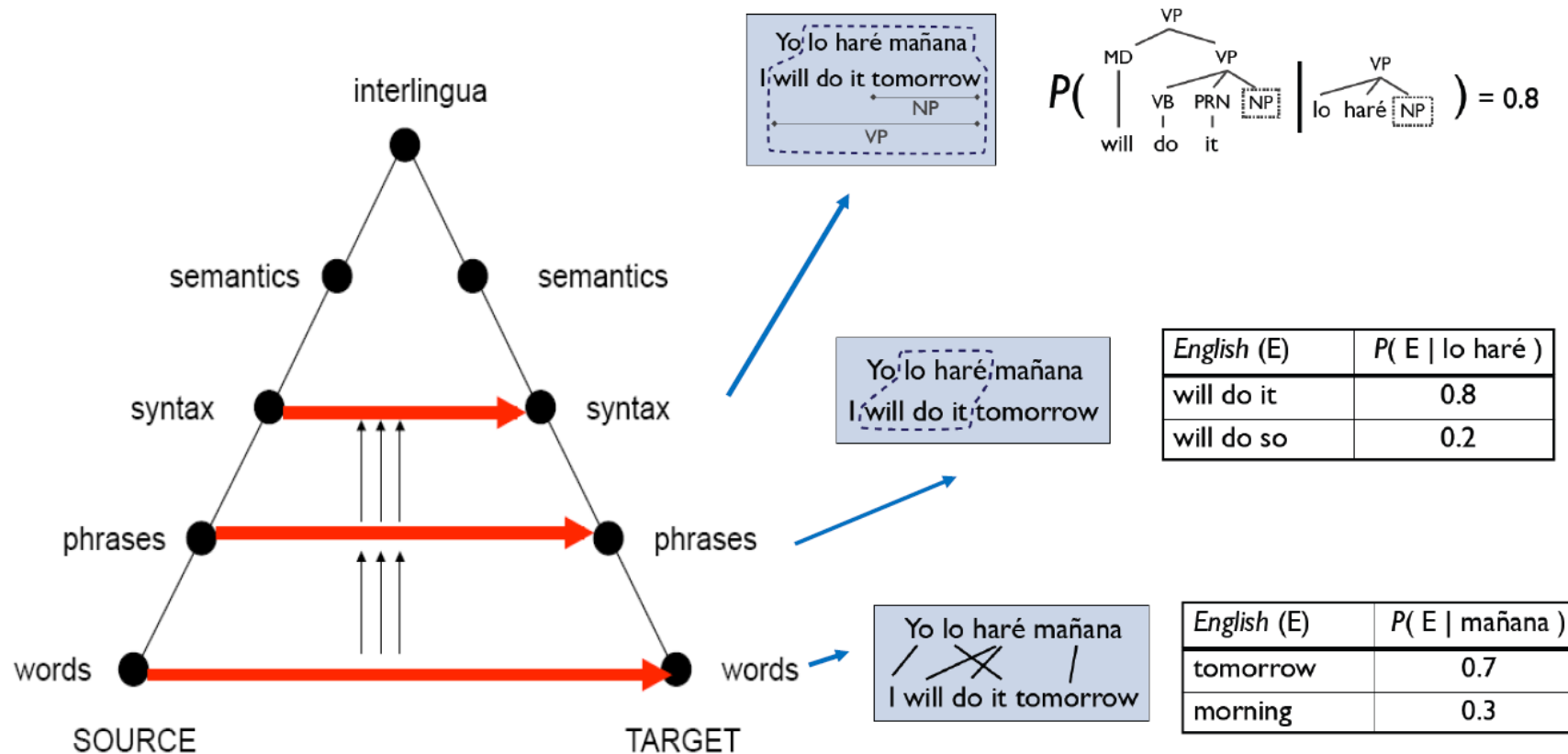
机器翻译中的词汇差异

- 一种语言中的词可能找不到它在另一种语言中的对译词汇

— 例如：

- 法语中按照是否流入大海，将“河流”分为“Rivière”和“fleuve”
- 德语中的Schedenfraude (幸灾乐祸, feeling good about another's pain)
- 日语中的“Oyakoko” (親孝, filial piety)

翻译的层次



词汇层的翻译

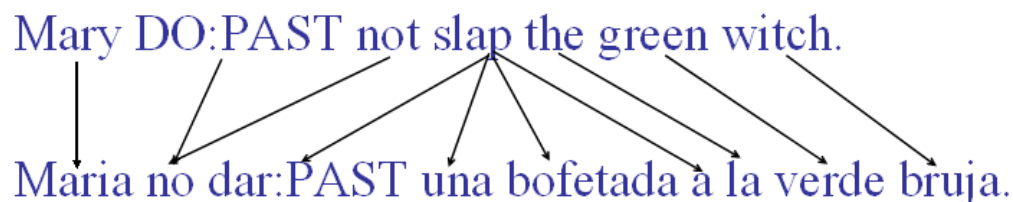
- 如何翻译一个词？
 - 查词典
 - Gap: 缺口;分歧;间隔;需加填补的不足、缺陷或空白
- 一对多的翻译：
 - 一种翻译较另一些翻译更加常见
 - 例如：gap被翻译为“分歧”的情况较多
 - 另一些比较少见
 - 例如：*a gap in the market*中gap被翻译为“脱销”

词汇层的翻译

- 形态分析:

- Mary didn't slap the green witch. →
Mary DO:PAST not slap the green witch.

- 词汇翻译:



- 词汇重排序:

- Maria no dar:PAST una bofetada a la bruja verde.

- 形态变换:

- Maria no dió una bofetada a la bruja verde.

词汇层翻译的问题

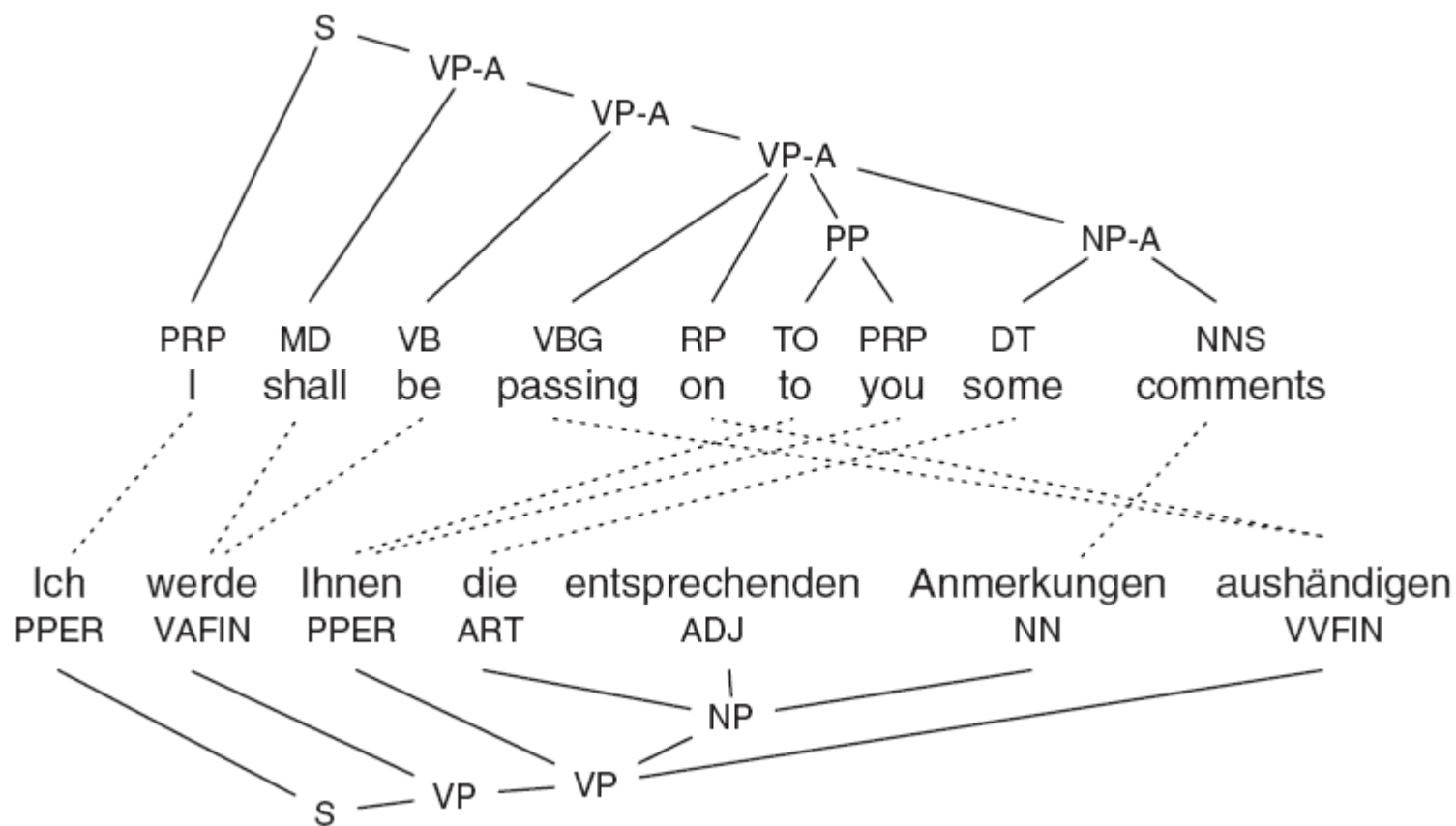
- 缺少对源语言的分析，可能会导致：
 - 很难捕获长距离的排序信息（不能较好的处理动态词序调整，例如需要将SVO结构翻译到SOV结构时）
 - 英语：Sources said that IBM bought Lotus yesterday
 - 日语：Sources yesterday IBM Lotus bought that said
 - 对词的语法角色不做歧义消解
 - 例如：that 可以是补语或定语，这两种情况下的翻译很不一样
 - They said that ...
 - They like that ice-cream

语法层的翻译

- 语法层的翻译将一种语言的句法树映射到另一种语言的句法树
 - 英语 \rightarrow 日语:
 - $VP \rightarrow V NP \Rightarrow VP \rightarrow NP V$
 - $PP \rightarrow P NP \Rightarrow PP \rightarrow NP P$
- 三个阶段:
 - 分析：对源语言句子做句法分析
 - 转换：将源语言中的句法树转换到目标语言中
 - 生成：采用目标语言中的句法树生成一个句子

语法层的翻译

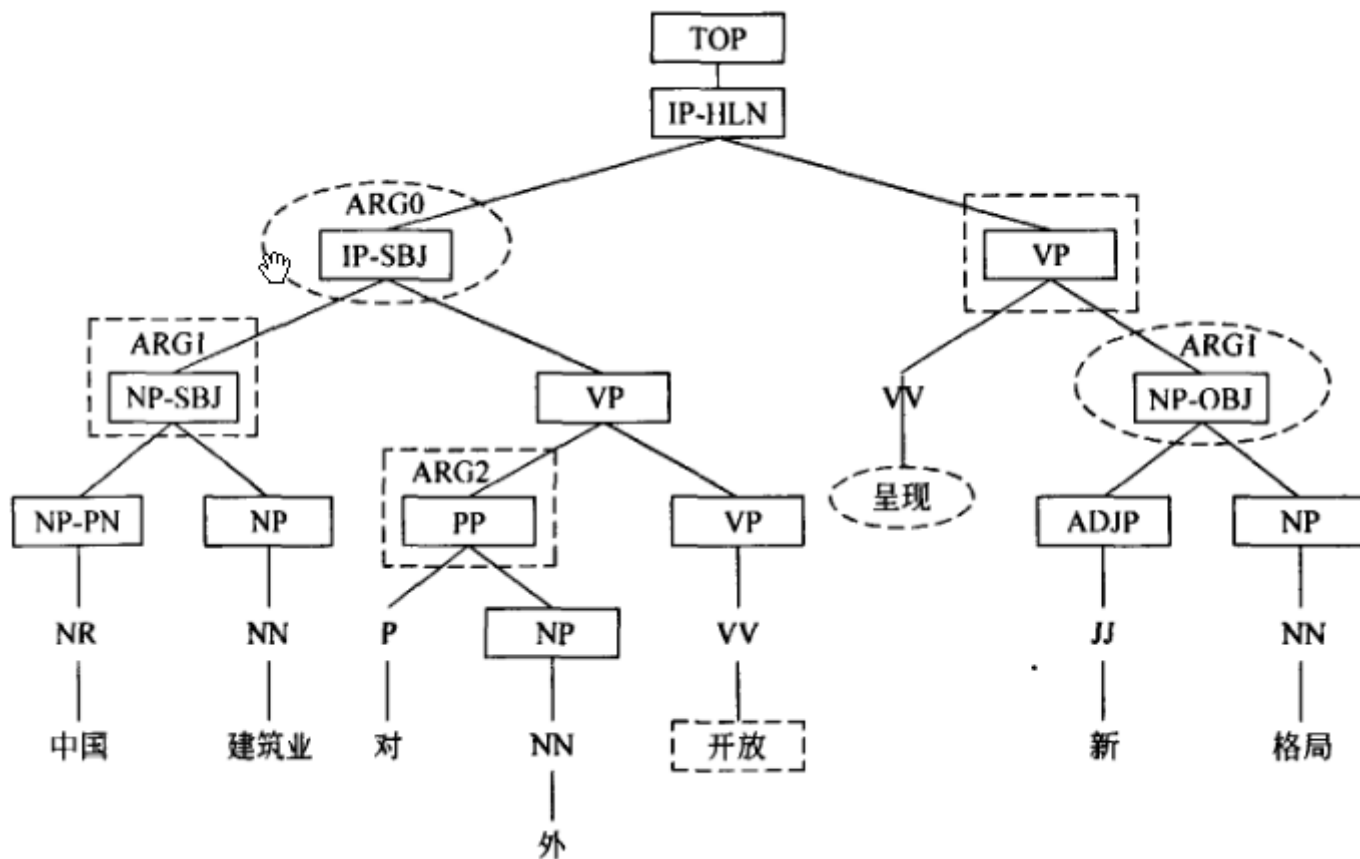
- 例子：一个词汇对齐的German-English句子树



语义层的翻译

- 一些结构歧义的分析需要语义信息
 - 介词短语歧义：
 - *Jim washes the dishes and watches television with Jane.*
- 一些句子的语义依赖于句子的结构：
 - 英语→汉语
 - $VP \rightarrow V PP[+benefactor] \Rightarrow VP \rightarrow PP[+benefactor] V$
 - She makes three meals a day for children \Rightarrow 她为孩子们做一日三餐
- 语义分析之语义角色分析
 - 句子中谓词所支配的语义角色：
 - 主体论元：施事、感事、经事、致事、主事；
 - 客体论元：受事、与事、对象、系事；
 - 凭借论元：工具、材料、方式、原因、目的；
 - 环境论元：时间、处所、源点、终点、路径、范围、量幅

语义层的翻译



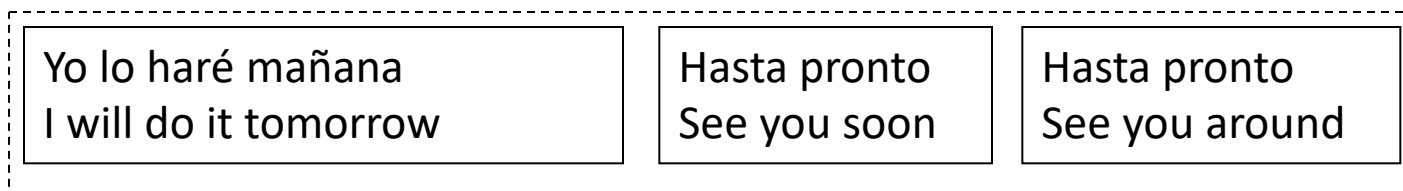
例子：一个标注了语义角色的句法树

统计机器翻译

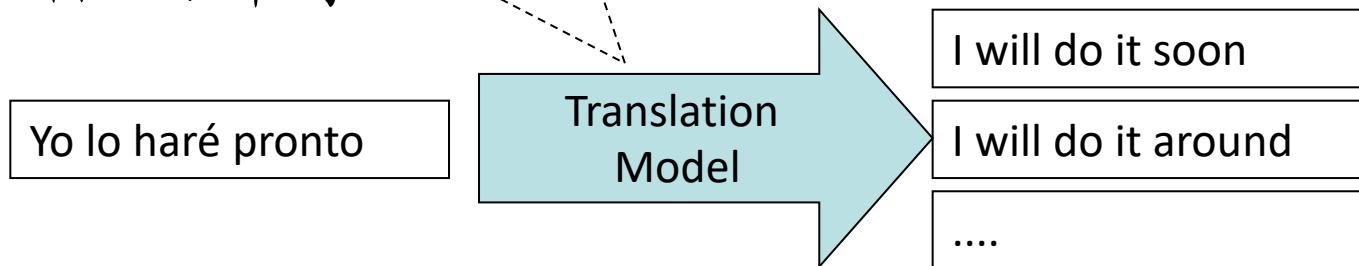
- 手工编制一套双语词典及翻译规则十分困难
- 往往需要从大量的平行语料 (parallel corpus) 或双语语料获取翻译知识
- 基于语料的机器翻译：
 - Translational English Corpus (TEC): 千万词汇规模的从各国语言翻译成英语的文本
 - 首先根据一些对齐规则 (例如句子长度等) 进行句子级的对齐

统计机器翻译

- 对语言之间的关联性进行建模
- 句子对齐 (sentence alignment) 的平行语料:



- 机器翻译系统:



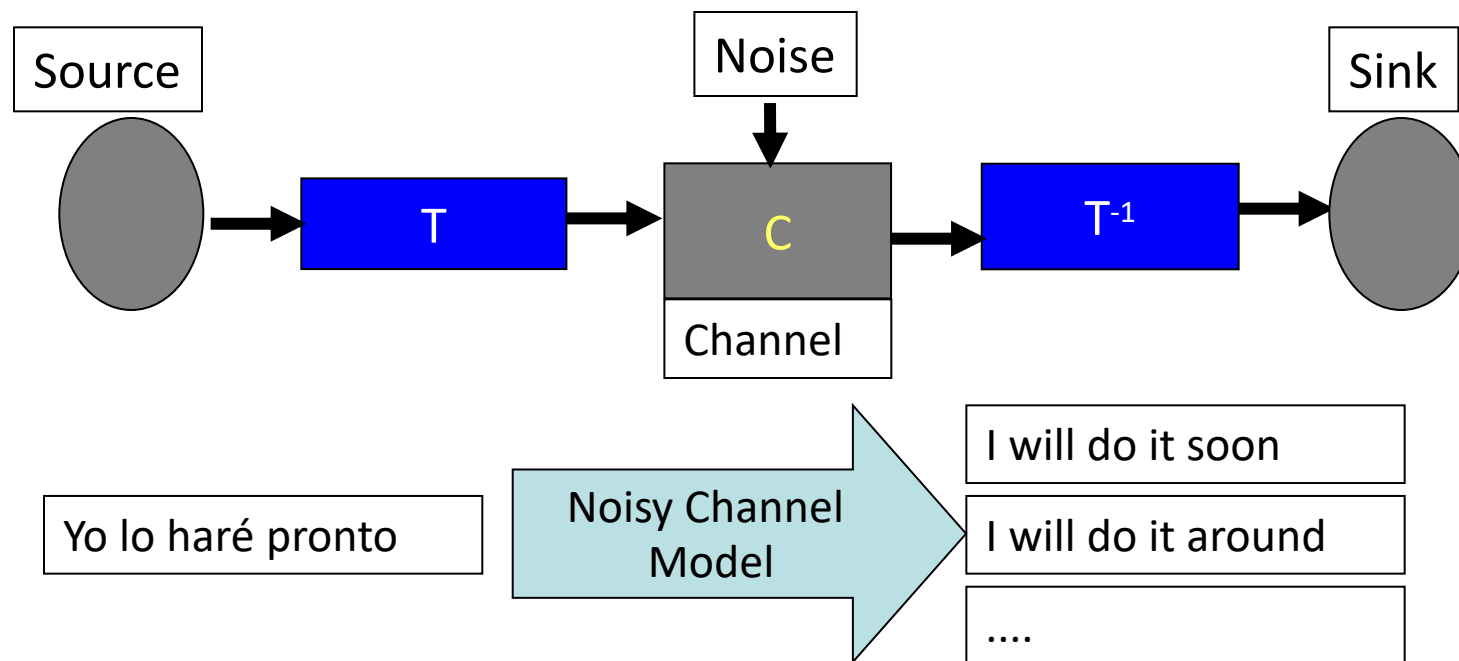
统计机器翻译

- 一个好的翻译应该具备的三个特点：
 - 严复：“译事三难：信、达、雅。”
 - 信(faithfulness): 正确且忠实地传达源语言包含的信息
 - 达(fluency): 语法结构正确、可读性好
- 翻译模型的目标：

$$T_{best} = \operatorname{argmax}_{T \in \text{Target}} \text{faithfulness}(T, S) \times \text{fluency}(T)$$

噪声信道模型

- 假设源语言句子是由某个目标语言句子经过噪声信道传播得到，使用Bayesian方法来找到最可能产生该源语言句子的目标语言句子



噪声信道模型

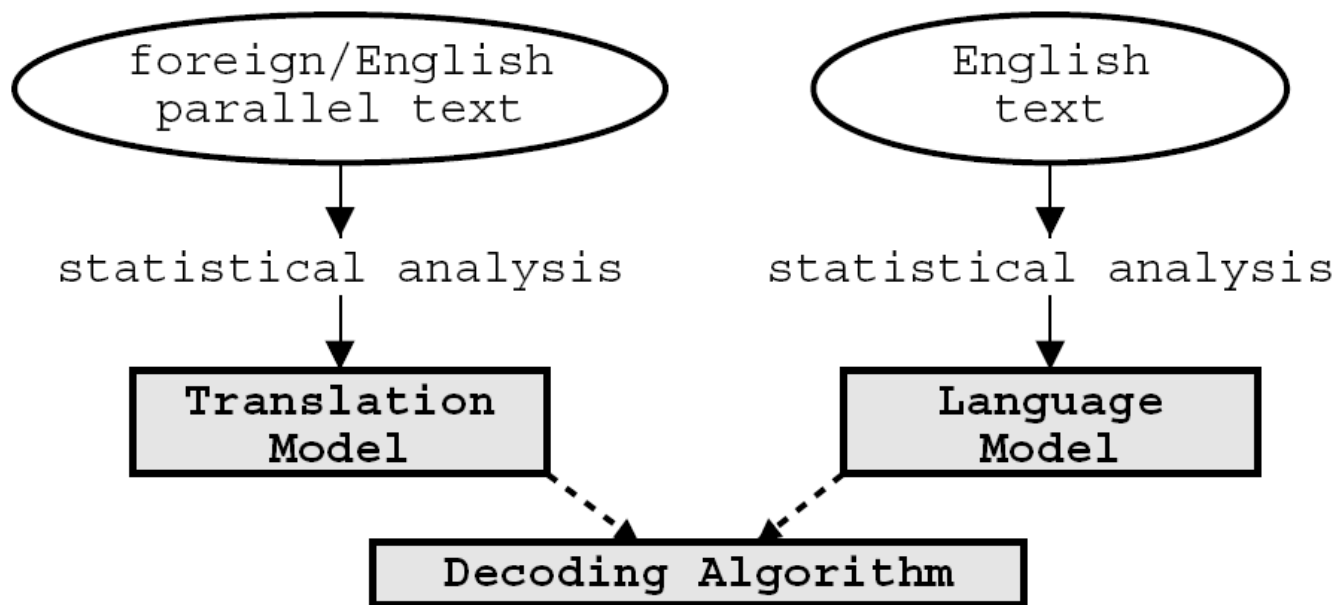
将源语言（其它语言）句子 $f=f_1, f_2, \dots, f_m$ 翻译到目标语言（英语）句子 $e=e_1, e_2, \dots, e_l$, 使得 $P(E | F)$ 最大化

$$\begin{aligned}\hat{e} &= \operatorname{argmax}_{e \in \text{English}} P(e | f) \\ &= \operatorname{argmax}_{e \in \text{English}} \frac{P(f | e)P(e)}{P(f)} \\ &= \operatorname{argmax}_{e \in \text{English}} \underbrace{P(f | e)}_{\text{Translation Model}} \underbrace{P(e)}_{\text{Language Model}}\end{aligned}$$

解码器（**decoder**）：提供一种给定 f 找到其最可能的翻译 e 的方法

噪声信道模型

- 构成：翻译模型、语言模型、解码器



语言模型 $p(e)$

- 可以采用 n -gram语言模型计算 $p(e)$
 - 可以通过目标语言 E 上一个无监督的单语语料训练得到
 - 已有T数量级的中英文web语料
- 也可以采用较复杂的PCFG语言模型来捕获长距离相依特性计算 $p(e)$

翻译模型 $p(f|e)$

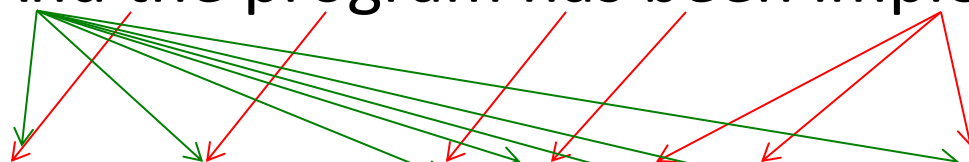
- 如何建模？
- IBM Model 1:
 - Brown *et al.* 在1993年提出，第一个真正意义上的统计机器翻译模型
 - 生成模型：将翻译过程分解为多个更小的步骤
 - 假设从 $e=e_1, e_2, \dots, e_l$ 产生 f 的生成过程如下：
 - 选择长度为 m 的句子 f : $f=f_1, f_2, \dots, f_m$
 - 选择一个一到多的对齐方式 A : $A=a_1, a_2, \dots, a_m$
 - 对于 f 中的词 f_j ，由 e 中相应的对齐词 e_{a_j} 生成

翻译模型 $p(f|e)$

- 几个符号：
 - 英语句子 e 包含 l 个词: $e_1 \dots e_l$
 - 外文句子 f 包含 m 个词: $f_1 \dots f_m$
- 对齐 (**alignment**): 一种对齐定义了每个外文词由哪 (些) 个英文词翻译过来
 - 即一个对齐 a 可以表示为 $\{a_1, \dots, a_m\}$, 其中, 对于 $j \in \{0, \dots, m\}$, $a_j \in \{0, \dots, l\}$
 - 存在 $(l+1)^m$ 中可能的对齐方式

翻译模型 $p(f|e)$

- e.g.,
 - $l = 6, m = 7$
 - $e =$ And the program has been implemented
 - $f =$ Le programme a ete mis en application
- 其中一种对齐为：
 - $\{2, 3, 4, 5, 6, 6, 6\}$
- 另一种对齐：
 - $\{1, 1, 1, 1, 1, 1, 1\}$



翻译模型 $p(f|e)$

- 目标式可以表示为：

$$p(f|e, m) = \sum_{a \in A} p(f, a|e, m)$$

– 其中A是所有可能对齐方式的集合

- 由链式法则可得：

$$p(f, a|e, m) = \underbrace{p(a|e, m)}_{\text{alignment}} \underbrace{p(f|a, e, m)}_{\text{word generation}}$$

IBM Model 1: 对齐概率

- 首先, 估计 $p(a|e, m)$
- IBM model 1假设所有的对齐方式具有相同的概率, 即:

$$p(a | e, m) = \frac{1}{(l + 1)^m}$$

IBM Model 1: 翻译概率

- 然后, 估计 $p(f \mid a, e, m)$

- IBM model 1 中:

$$p(f \mid a, e, m) = \prod_{j=1}^m t(f_j \mid e_{a_j})$$

- $t(f_j \mid e_{a_j})$ 表示英文词 e_{a_j} 翻译为外文词 f_j 的概率

IBM Model 1: 翻译概率

- e.g., $l = 6, m = 7$
 - e = And the program has been implemented
 - f = Le programme a ete mis en application
 - $a = \{2, 3, 4, 5, 6, 6, 6\}$
- $p(f \mid a, e, 7) = t(\text{Le} \mid \text{the})$
 - $X \ t(\text{programme} \mid \text{program})$
 - $X \ t(a \mid \text{has})$
 - $X \ t(\text{ete} \mid \text{been})$
 - $X \ t(\text{mis} \mid \text{implemented})$
 - $X \ t(\text{en} \mid \text{implemented})$
 - $X \ t(\text{application} \mid \text{implemented})$

IBM Model 1: 生成过程

- 从英语句子 e 生成一个外文句子 f :
 - Step 1: 依据概率 $1/(l+1)^m$ 挑选一种对齐方式
 - Step 2: 依据下列概率选择外文句子

$$p(f | a, e, m) = \prod_{j=1}^m t(f_j | e_{a_j})$$

- 进而得到:

$$p(f, a | e, m) = p(a | e, m) \times p(f | a, e, m) = \frac{1}{(l+1)^m} \prod_{j=1}^m t(f_j | e_{a_j})$$

- 最后得到:

$$p(f | e, m) = \sum_{a \in A} p(f, a | e, m)$$

IBM Model 1: 得到最优对齐

- 对于给定的<f, e>对，可以计算某种对齐a的概率：

$$\begin{aligned} p(a | f, e, m) &= \frac{p(f, a | e, m)}{p(f | e, m)} \\ &= \frac{p(a | e, m) p(f | a, e, m)}{\sum_{a \in A} p(f, a | e, m)} \end{aligned}$$

- 进而，给定<f, e>对，可以计算其最可能的对齐方式：

$$a^* = \arg \max_a p(a | f, e, m)$$

- 如今IBM模型几乎不直接用于机器翻译，而多用于求解最优对齐

IBM Model 2

- 一个区别：引入对齐时的扭曲系数（distortion parameters）

- $q(i | j, l, m) =$

给定e和f的长度分别为l和m时，第j个外文词和第i个英文词对齐的概率

- 定义：

$$p(a | e, m) = \prod_{j=1}^m q(a_j | j, l, m)$$

其中 $a = \{a_1, \dots, a_m\}$

- 则：

$$p(f, a | e, m) = \prod_{i=1}^m q(a_j | j, l, m) t(f_j | e_{a_j})$$

IBM Model 2

- E.g.,
 - $l = 6$
 - $m = 7$
 - e = And the program has been implemented
 - f = Le programme a ete mis en application
 - $a = \{2, 3, 4, 5, 6, 6, 6\}$

$p(a \mid e, 7) = q(2 \mid 1, 6, 7)$ $p(f \mid a, e, 7) = t(\text{Le} \mid \text{the})$

X $q(3 \mid 2, 6, 7)$

X $t(\text{programme} \mid \text{program})$

X $q(4 \mid 3, 6, 7)$

X $t(a \mid \text{has})$

X $q(5 \mid 4, 6, 7)$

X $t(\text{ete} \mid \text{been})$

X $q(6 \mid 5, 6, 7)$

X $t(\text{mis} \mid \text{implemented})$

X $q(6 \mid 6, 6, 7)$

X $t(\text{en} \mid \text{implemented})$

X $q(6 \mid 7, 6, 7)$

X $t(\text{application} \mid \text{implemented})$

IBM Model 2: 生成过程

- 从英文句子 e 生成外文句子 f 的过程:
 - Step 1: 依据如下概率选择一种对齐方式: $a = \{a_1, a_2, \dots, a_m\}$

$$\prod_{j=1}^m q(a_j | j, l, m)$$

- Step 2: 依据如下概率选择一个外文句子 f :

$$p(f | a, e, m) = \prod_{j=1}^m t(f_j | e_{a_j})$$

- 进而得到:

$$p(f, a | e, m) = p(a | e, m)p(f | a, e, m) = \prod_{j=1}^m q(a_j | j, l, m)t(f_j | e_{a_j})$$

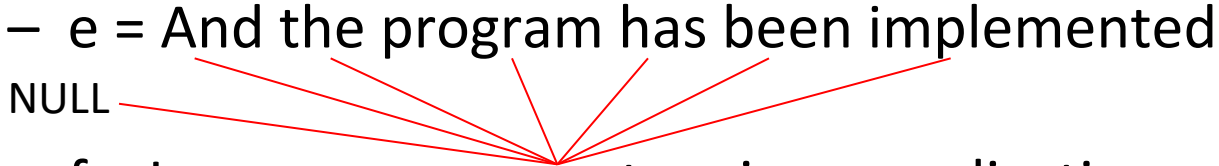
- 最后得到:

$$p(f | e, m) = \sum_{a \in A} p(f, a | e, m)$$

IBM Model 2: 得到最优对齐

- 如果已经得到参数 q 和 t , 则对于每个句对 $e_1, e_2, \dots, e_l, f_1, f_2, \dots, f_m$, 其最优对齐 a_j ($j = 1, \dots, m$) 为:

$$a_j = \arg \max_{a \in \{0 \dots l\}} q(a|j, l, m) \times t(f_j|e_a)$$

- E.g.,
 - e = And the program has been implemented
NULL
 - f = Le programme **a** ete mis en application
- 

考察 $f_3=a$ 的最优对齐时，计算：

$a \leftarrow \text{NULL}: q(0|3, 6, 7) * t(a|\text{NULL})$

$a \leftarrow \text{the}: q(1|3, 6, 7) * t(a|\text{and})$

$a \leftarrow \text{program}: q(2|3, 6, 7) * t(a|\text{program})$

....

$a \leftarrow \text{implemented}: q(6|3, 6, 7) * t(a|\text{implemented})$

从中选取一个概率最大的对齐方式

参数估计问题

- 输入: $(e^{(k)}, f^{(k)})$, $k = 1, \dots, n$, $e^{(k)}$ 表示第 k 个英文句子, $f^{(k)}$ 表示第 k 个外文句子
- 输出: 参数 $t(f|e)$ 和 $q(i|j, l, m)$
- 一个挑战: 只有双语句子对齐语料, 例如:
 - $e^{(100)} = \text{And the program has been implemented}$
 - $f^{(100)} = \text{Le programme a ete mis en application}$

参数估计：极大似然估计

- 如果训练语料包含词-词的对齐信息
- e.g.,
 - $e^{(100)}$ = And the program has been implemented
 - $f^{(100)}$ = Le programme a ete mis en application
 - $a^{(100)} = \{2, 3, 4, 5, 6, 6, 6\}$
- 即：训练数据为
 - $(e^{(k)}, f^{(k)}, a^{(k)})$, $k = 1, \dots, n$, $e^{(k)}$ 表示第k个英文句子, $f^{(k)}$ 表示第k个外文句子, $a^{(k)}$ 表示第k组句子中的词对齐
- 采用极大似然估计法：

$$t_{ML}(f|e) = \frac{\text{Count}(e, f)}{\text{Count}(e)} \quad q_{ML}(j|i, l, m) = \frac{\text{Count}(j|i, l, m)}{\text{Count}(i, l, m)}$$

Input: A training corpus $(f^{(k)}, e^{(k)}, a^{(k)})$ for $k = 1 \dots n$, where $f^{(k)} = f_1^{(k)} \dots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \dots e_{l_k}^{(k)}$, $a^{(k)} = a_1^{(k)} \dots a_{m_k}^{(k)}$.

Algorithm:

- ▶ Set all counts $c(\dots) = 0$
- ▶ For $k = 1 \dots n$
 - ▶ For $i = 1 \dots m_k$, For $j = 0 \dots l_k$,

$$c(e_j^{(k)}, f_i^{(k)}) \leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k, i, j)$$

$$c(e_j^{(k)}) \leftarrow c(e_j^{(k)}) + \delta(k, i, j)$$

$$c(j|i, l, m) \leftarrow c(j|i, l, m) + \delta(k, i, j)$$

$$c(i, l, m) \leftarrow c(i, l, m) + \delta(k, i, j)$$

where $\delta(k, i, j) = 1$ if $a_i^{(k)} = j$, 0 otherwise.

Output: $t_{ML}(f|e) = \frac{c(e, f)}{c(e)}$, $q_{ML}(j|i, l, m) = \frac{c(j|i, l, m)}{c(i, l, m)}$

k: 训练样本序号, i: $f^{(k)}$ 中的第i个词, j: $e^{(k)}$ 中的第j个词

参数估计：EM算法

- 如果训练语料仅包含： $(e^{(k)}, f^{(k)})$, $k = 1, \dots, n$, $e^{(k)}$ 表示第k个英文句子， $f^{(k)}$ 表示第k个外文句子
- 算法过程与已知词对齐时类似
- 两个关键不同：
 - 通过迭代计算模型参数 q 和 t
 - 从一个初始（例如随机选取的） q 和 t 出发
 - 每次迭代时根据训练数据和当时的 q 、 t 计算“counts”
 - 依据当前的“counts”重新估计 q 和 t
 - 每次迭代时，依据下式计算 $\delta(k, i, j)$

$$\delta(k, i, j) = \frac{q(j|i, l_k, m_k) t(f_i^{(k)} | e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i, l_k, m_k) t(f_i^{(k)} | e_j^{(k)})}$$

k : 训练样本序号, i : $f^{(k)}$ 中的第 i 个词, j : $e^{(k)}$ 中的第 j 个词

参数估计：EM算法

Input: A training corpus $(f^{(k)}, e^{(k)})$ for $k = 1 \dots n$, where $f^{(k)} = f_1^{(k)} \dots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \dots e_{l_k}^{(k)}$.

Initialization: Initialize $t(f|e)$ and $q(j|i, l, m)$ parameters (e.g., to random values).

参数估计：EM算法

For $s = 1 \dots S$

- ▶ Set all counts $c(\dots) = 0$
- ▶ For $k = 1 \dots n$
 - ▶ For $i = 1 \dots m_k$, For $j = 0 \dots l_k$

$$c(e_j^{(k)}, f_i^{(k)}) \leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k, i, j)$$

$$c(e_j^{(k)}) \leftarrow c(e_j^{(k)}) + \delta(k, i, j)$$

$$c(j|i, l, m) \leftarrow c(j|i, l, m) + \delta(k, i, j)$$

$$c(i, l, m) \leftarrow c(i, l, m) + \delta(k, i, j)$$

where

$$\delta(k, i, j) = \frac{q(j|i, l_k, m_k) t(f_i^{(k)} | e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i, l_k, m_k) t(f_i^{(k)} | e_j^{(k)})}$$

- ▶ Recalculate the parameters:

$$t(f|e) = \frac{c(e, f)}{c(e)} \quad q(j|i, l, m) = \frac{c(j|i, l, m)}{c(i, l, m)}$$

k: 训练样本序号, i: $f^{(k)}$ 中的第 i 个词, j: $e^{(k)}$ 中的第 j 个词

参数估计：EM算法

$$\delta(k, i, j) = \frac{q(j|i, l_k, m_k) t(f_i^{(k)} | e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i, l_k, m_k) t(f_i^{(k)} | e_j^{(k)})}$$

$e^{(100)}$ = And the program has been implemented

$f^{(100)}$ = Le programme **a** ete mis en application

$$\Delta = q(0|3, 6, 7) * t(a|NULL) + q(1|3, 6, 7) * t(a|and) + \dots$$

$$\delta(100, 3, 0) = q(0|3, 6, 7) * t(a|NULL) / \Delta$$

$$\delta(100, 3, 1) = q(1|3, 6, 7) * t(a|and) / \Delta$$

$$\delta(100, 3, 2) = q(2|3, 6, 7) * t(a|the) / \Delta$$

....

$$\delta(100, 3, 6) = q(6|3, 6, 7) * t(a|implemented) / \Delta$$

事实上： $\delta(k, i, j) = p(a_i^{(k)} = j \mid e^{(k)}, f^{(k)}; q, t)$

参数估计：EM算法

- 算法验证：

- 训练语料： $(e^{(k)}, f^{(k)})$, $k = 1, \dots, n$, $e^{(k)}$ 表示第 k 个英文句子, $f^{(k)}$ 表示第 k 个外文句子
- Log似然函数：

$$L(t, q) = \sum_{k=1}^n \log p(f^{(k)} | e^{(k)}) = \sum_{k=1}^n \log \sum_a p(f^{(k)}, a | e^{(k)})$$

- 极大似然估计：

$$\arg \max_{t, q} L(t, q)$$

- EM算法可收敛于log似然函数的局部最优值

总结

- IBM翻译模型的主要思路：
 - 对齐参数: a
 - 翻译概率: e.g., $t(\text{chien}|\text{dog})$
 - 扭曲因子: e.g., $q(2|1, 6, 7)$
- EM算法: 迭代计算参数 q 和 t
- 由训练数据训练得到以上参数后, 可以据此计算最可能的对齐 a^*

- Next lecture: neural machine translation