可参考 http://www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/ibm12.pdf

内容摘自 http://blog.csdn.net/dark_scope/article/details/8774000

1.Challenge

首先还是看看挑战吧，万变不离其宗，一切nlp的问题基本上都离不开语言的Ambiguity这个词，相同词汇在不同语句中的词性，意义都可能会不同。这还是在一门语言中，机器翻译涉及到两门语言，其问题更甚，比如同一个词可能会有很多种不同的翻译方法。

除此之外，不同语言的句子构成结构是不同的，比如下面是英语和日语的构句形式：

English word order is *subject – verb – object*

Japanese word order is *subject – object – verb*

2.Direct Machine Translation

最简单的翻译方法就是直接翻译，如你所想，一个词一个词地翻译，基本上是不懂英语的人的水平，翻译出来的结果可想而知。不过这也是机器翻译的鼻祖了吧。对于一个词的翻译你得用很多歌if else来写，这样一个模型的缺点当然就很多啦，要靠人工来写if else，这是相当费时费力还不讨好的事情，而且没有考虑单词的意义。

除此之外还有一些“经典”的翻译模型，也就是过时的，落后的，比如通过分析两种语言不同的构句结构来调整语法树，从而得到翻译的句子。从结果来讲都没有statistic的模型来得有效

3.The Noisy Channel Model

The Noisy Channel Model有两个部分组成：

$p(e)$ **the language model**

$p(f \mid e)$ **the translation model**

我们用e代表英语，f代表法语，因为IBM的模型是在这两门语言上进行实验的。这里是由法语翻译为英语（这个一定要时刻记着啊！）

通过bayes的概率论的转换我们就可以得到：

$$p(e \mid f) = \frac{p(e,f)}{p(f)} = \frac{p(e)p(f \mid e)}{\sum_e p(e)p(f \mid e)}$$

$$\text{argmax}_e p(e \mid f) = \text{argmax}_e p(e)p(f \mid e)$$

我们的翻译结果就是argmax所得到的e。为什么分母是可以不用考虑的呢，因为这个分母是个固定的数，就是P(f)

注意：语言模型p(e)和我们之前定义的是一样的。而翻译模型我们会从很多句一一对应的英法语句中训练学习到。这个模型是IBM两个模型的基础。

在这里插一句，先给出一些基础的定义：

Definition 1 (IBM Model 2) *An IBM-M2 model consists of a finite set $\mathcal{E}$ of English words, a set $\mathcal{F}$ of French words, and integers M and L specifying the maximum length of French and English sentences respectively. The parameters of the model are as follows:*

后面会用e,f,m,l表示。

英文原文中是先讲的IBM M2，但是我也觉得作者先写M1,再写出来M2，这样更让人好接受一些。

4.IBM 1 Model

IBM 1模型引入了一个很重要的东西：Alignments，也就是词汇之间是如何关联的。很明显不同语言中同一个意思的句子词汇之间肯定会联系起来，这是翻译的基础。所谓的关联就是句子中词与词的对应关系。

比如说有如下两个句子：

$$e = \text{And the program has been implemented}$$

$$f = \text{Le programme a ete mis en application}$$

l=6表示英语句子的长度，m=7表示法语句子的长度。

一个alignment就是一个长度为m的序列，将f中的每一个词映射到e中，为了完善，我们会为e额外增加一个序号0，用来表示映射为空。（为什么需要这样一个null的值呢。这个我们后面再说这个问题，在概率模型中会使用到这个东西）。

比如说一个allignment：

$$\{2, 3, 4, 5, 6, 6, 6\}$$

就表示如下映射：

e = And the program has been implemented

f = Le programme a ete mis en application

多个词可以映射到同一个词。

有了Alignments之后，我们就可以改变我的模型了，我们有如下定义

$$p(f, a \mid e, m) = p(a \mid e, m) p(f \mid a, e, m)$$

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

模型的基本框架就是上面的这个公式。

其中a表示一个alignment，就是一个对应关系的数组。上面的那个2345666。

在这个过程中我们其实还可以得到很多有用的东西，比如说“最大可能的Alignment”，给你两个句子，必然有一个最大可能的映射。显然是概率中的问题，对于e和m，最常出现的a序列，就是最大可能的Alignment。

给定一个f,e，当然也就知道了l和m。我们有：

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

值得一提的是，IBM Model 1虽然现在没有用于翻译，但它仍被用来寻找“最大可能的Alignment”

在M1 中：

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

所有alignment都是一样的，这是一个很强的假设，但同时也是所有事情的开始~~~

这是我们总的模型的右边部分，左边部分是上式，好，那我们再来看看右边的式子是什么：

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

t是什么东西呢，t代表的是从英文句子的e单词翻译到法语句子中单词f的概率，至于这个概率咋求，别急，之后会看到，原文的内容是：

- *$t(f|e)$ for any $f \in \mathcal{F}$, $e \in \mathcal{E} \cup \{NULL\}$. The parameter $t(f|e)$ can be interpreted as the conditional probability of generating French word f from English word e.*

举个栗子：

e = And the program has been implemented

f = Le programme a ete mis en application

- $a = \{2, 3, 4, 5, 6, 6, 6\}$

$$\begin{aligned} p(f \mid a, e) = \; & t(Le \mid the) \times \\ & t(programme \mid program) \times \\ & t(a \mid has) \times \\ & t(ete \mid been) \times \\ & t(mis \mid implemented) \times \\ & t(en \mid implemented) \times \\ & t(application \mid implemented) \end{aligned}$$

综上，IBM 1 Model：

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

至于t到底怎么求，这个我们在后面会讲到。

5.IBM Model 2

我们先介绍一下Model 2中引入的新的元素：

- *$q(j|i, l, m)$ for any $l \in \{1 \ldots L\}$, $m \in \{1 \ldots M\}$, $i \in \{1 \ldots m\}$, $j \in \{0 \ldots l\}$. The parameter $q(j|i, l, m)$ can be interpreted as the probability of alignment variable a_i taking the value j, conditioned on the lengths l and m of the English and French sentences.*

q(j|i,l,m)表示的是：给定l,m，a[i] = j的概率，表示就是i这个。这个概率怎么求？也在后面。

然后有下面这个式子，下面表示成q(aj | j,l,m)只不过是把上面的i换成了j。

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

注意这就是Model 1中不同的地方了，Model 1中所有alignment的该项值都是相等的，这里是不等的

那么总的模型就是：

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

这就是Model 2中和1的差别

6.EM Training of Models 1 and 2

那么怎么计算这两个模型呢

一、 我们首先介绍alignment已经存在的情况，也就是训练集包括e,f,a，就是在训练阶段给了英语句子翻译成什么法语句子，还有每个发语词对应英语词的对应关系。

对t和q做如下统计计算：

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

具体的伪代码如下：

Algorithm:

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

$$\begin{aligned} c(e_j^{(k)}, f_i^{(k)}) &\leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k,i,j) \\ c(e_j^{(k)}) &\leftarrow c(e_j^{(k)}) + \delta(k,i,j) \\ c(j|i,l,m) &\leftarrow c(j|i,l,m) + \delta(k,i,j) \\ c(i,l,m) &\leftarrow c(i,l,m) + \delta(k,i,j) \end{aligned}$$

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表示的是多少组句子。

二、这种情况相对比较简单，我们再来看训练集中不给出a的情况：

For $s = 1 \ldots S$

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

$$
\begin{aligned}
c(e_j^{(k)}, f_i^{(k)}) &\leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k,i,j) \\
c(e_j^{(k)}) &\leftarrow c(e_j^{(k)}) + \delta(k,i,j) \\
c(j|i,l,m) &\leftarrow c(j|i,l,m) + \delta(k,i,j) \\
c(i,l,m) &\leftarrow c(i,l,m) + \delta(k,i,j)
\end{aligned}
$$

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:

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

其思想是EM算法的思想，通过下式计算 $\delta(k,i,j)$，来实现逐步地趋近于最优值，注意每次都会重新计算t和q的值

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

EM的算法在前两篇文章里有说过。为啥会使用上面的这个公式呢。。。。我也不知道，文章中就是这么用的，这个公式可以适用于EM算法，每个Count都是期望E，再用期望去算

再者，上面的训练是通用的，如果对于M1的话，只要

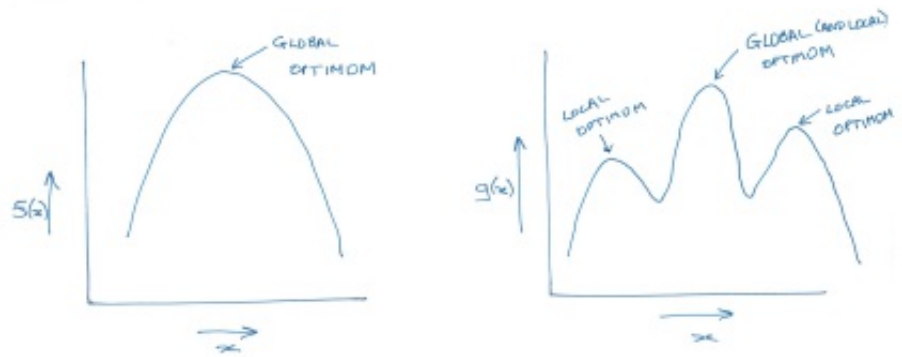
$$
\delta(k,i,j) = \frac{q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}
$$

with

$$
\delta(k,i,j) = \frac{\frac{1}{(l^{(k)}+1)}t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k}\frac{1}{(l^{(k)}+1)}t(f_i^{(k)}|e_j^{(k)})} = \frac{t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} t(f_i^{(k)}|e_j^{(k)})}
$$

7.一些改进

首先，对于Model2来说，t和q都是变化的，也就是可能不是单调的凸函数，所以再使用em算法进行求解的时候会可能出现局部最小值。



但是如果在M1中，q有个强假设，所以只有t是变化的，所以是个凸函数，所以不管t的初始值设置成啥，都会最终趋近与全局最大值。所以

实践表明，先用Model1去计算出来t参数，然后用这个t参数，和随机出来的q去初始化M2的训练，效果会更好！

最后：为啥要M1，M2呢，能干啥呢？

对于每个给定的f，a，e。我们都来计算

$$p(f|e) = \sum_a p(f, a|e)$$

然后我们解出来：

$$\arg\max_e p(e)p(f|e)$$

这个过程叫decoding，其实这个复杂度是很大的，但是有很多近似的方法（虽然我还不知道呢）。

所以M1 和M2其实不是一个好的模型（瞎了，学这么久还不是好的）。但是吧，为啥还要有这个的存在，因为这个里面求出来的t和a（alignment）是很重要的，在其他的翻译系统中，可以派上很大的用途。原英文中有简单的说明。反正就是这句话的意思。