IBM model1的推导与实现

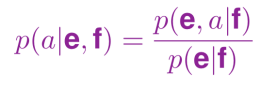
一．重要概念说明：

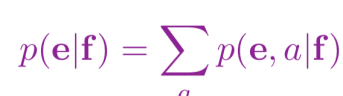
1. alignment:在平行文本中，我们将一种语言中的单词与另一种语言中的单词的对齐叫做alignment(eg:das->the,Haus->house,ist->is,klein->small)，alignment只是对应关系，并不是概率值
2. alignment function:将i位置的一种语言单词，映射到另一语言单词位置j的函数叫alignment function(a:i->j).在python中通常用dict来表示对应关系，如(das,Haus,ist,klein) : (the,house,is,small)
3. IBM model1:该模型是以统计alinment下语言1翻译为语言2的概率。IBM Model 1 采用最简单的假设：假设每个alignment所发生的概率相同。

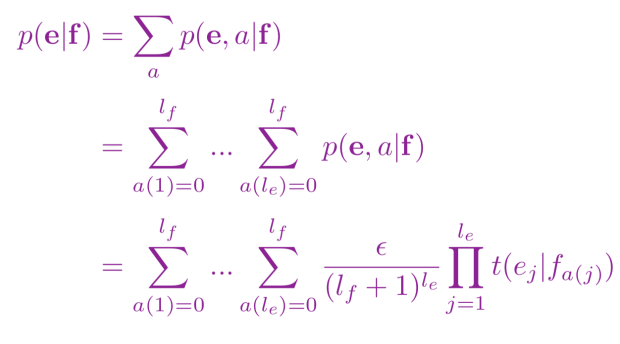
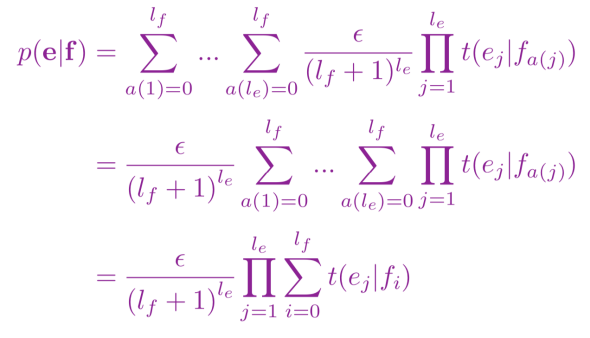
以法文翻译为英文为例，如果法文单词有I个，对应翻译的英文单词有J个，则有I^J种（^表示指数符号，I的J次方）可能的对应关系，实际上有些法文是没有对应英文翻译的，（法文我不太了解，但是中文的话，以”了”,”吗”,”的”等字，其实翻译不出来英文的）算上没有对应关系的可能，则有(I+1)^J种可能。以下是model1中重要的元素声明：

* 1. 外来句子（法文）以f=(f1,f2,…,f\_lf)表示，长度为lf
  2. 英文句子（英文）以e=(e1,e2,…,e\_le)表示，长度为le
  3. 目标函数：P(a|e,f)，表示通过法语和英语为条件，找出其alignment。
  4. 参数估计：EM算法，最大化期望算法，通过概率模型中寻找参数最大似然估计或者最大后验估计的算法，其中概率模型依赖于无法观测的隐性变量。最大期望算法经过两个步骤交替进行计算，第一步是计算期望（E），利用对隐藏变量的现有估计值，计算其最大似然估计值；第二步是最大化（M），最大化在E步上求得的最大似然值来计算参数的值。M步上找到的参数估计值被用于下一个E步计算中，这个过程不断交替进行，使参数不断接近最大似然值。

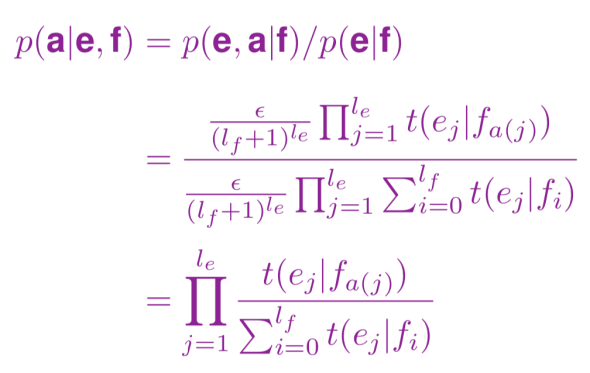
二．IBM model目标函数推导：

通过贝叶斯定理，可将函数变为如下形式，

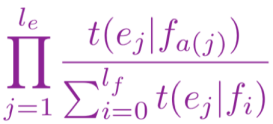
目标变为了计算P(e,a|f)和P(e|f)，先计算分母，分母可以写成左边全概率的形式，就写成了分子这样的累加形式。推导顺序从左往右看，最终得到的t(e\_j|f\_i)就是i位置法文对应j位置英文的alignment下，法文单词f\_i翻译为英文单词e\_i的概率，且每个位置上的概率都是参数的一部分。IMG_256 是常数项。

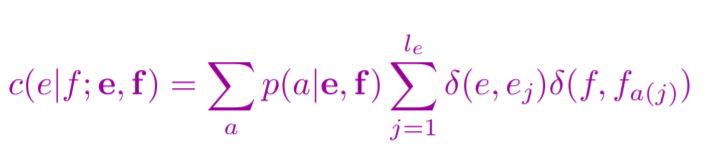
 

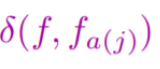
由此可将最终目标函数写成如下形式：

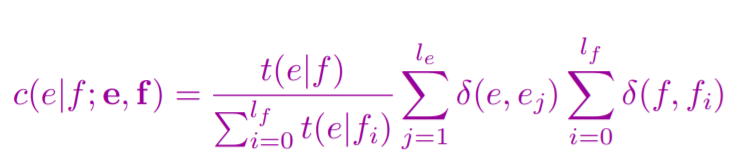


三．EM算法估计参数：

有了上述的目标函数的推导，我们明确知道了目标，就是计算。在求解的时候，不会把 alignment 的结果都加起來，而是求出概率最大的 alignment。因为我们有训练的平行语料，语料是由两种语言对应的翻译构成，但通常平行语料库中不会标记alignment，但是alignment又是必要的，这时就需要通过EM算法，计算t(e|f)，以得出最大的t(e|f)值。如何得到t(e|f)呢？在本例中是法文与英文的pair，我们可以通过统计平行语料中的所有源语言单词（上一节中以法文为例），与目标语言（上一节中的英文）对应的次数c(e|f)，以计算transition prob，作为整个模型的参数。我们先来看下c(e|f)的计算：



C(e|f;e,f)为在给定e的条件下，e链接到f的期望次数。大家看到C(e|f;e,f)可能有点懵逼，其实是这样的，整体表示的是c(e|f)，分号隔开的e,f表示的是相应的英文和法文的语料库。这个表示ej单词在句子E中的count数（期望次数）；表示fj单词在F句子中的count数（期望次数）。代入之前计算出的P(a|e,f)



这样我们就可以通过期望次数c(e|f)来更新每个参数了。

首先我们可以用uniform的方法初始化所有transition prob，然后通过EM算法，不断循环遍历训练平行语料库中的单词对应关系的计数来计算每个源语言单词对应语料中目标语言单词对应的概率，然后更新transition probs参数。先来看看EM算法流程：

1.E步骤：将模型应用于数据

- 隐藏模型的部分（此处：alignment）

- 使用模型，将概率分配给可能的值

2.M步骤：根据数据估算模型

- 将赋值作为事实

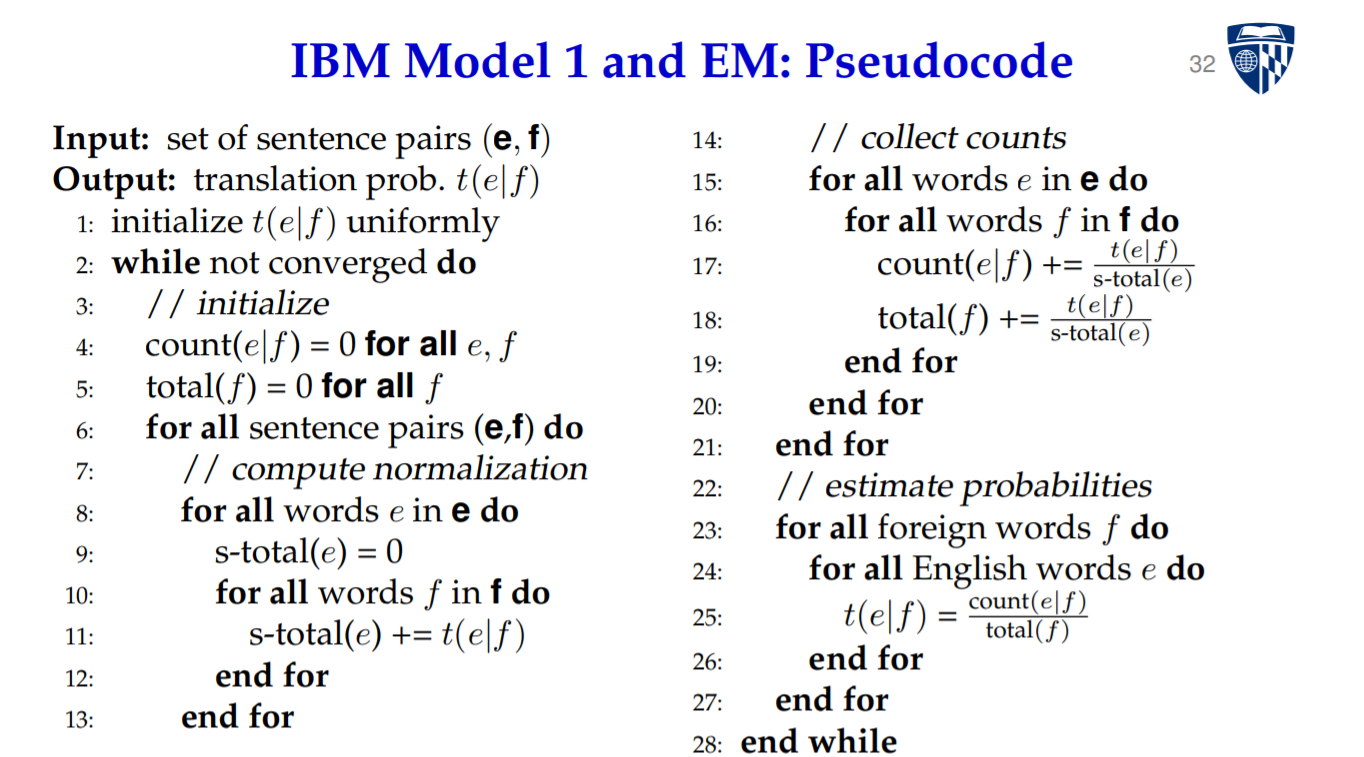
- 收集计数（按概率加权）

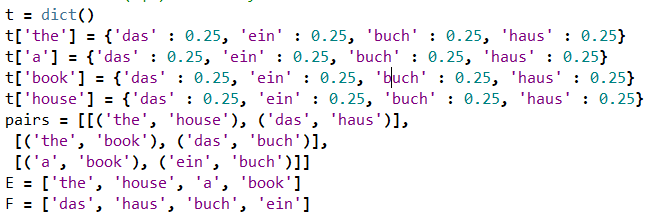
- 从计数中估算模型

3.迭代这些步骤直到收敛

四.实现建议

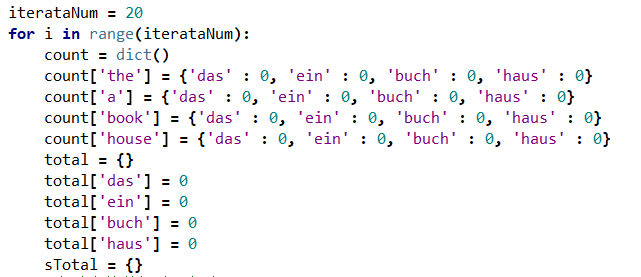
整个目标是通过EM算法学习IBM model中的transition prob参数，通过EM算法，可以在迭代统计平行语料的过程中，不断计算概率t(e|f)的似然概率，并更新到参数中，以达到学习整个参数中最大alignment对应最大概率的目的，我们看看lecture中给出的伪代码：

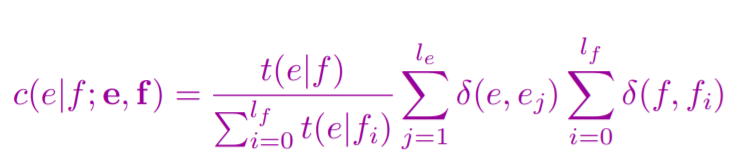


1. Initialize t(f|e)，伪代码对应1行：此步为初始化参数步骤，参数即源语言的每个单词与目标语言每个单词的transition prob(t(e|f))，初始化时可以使用uniform的方法。

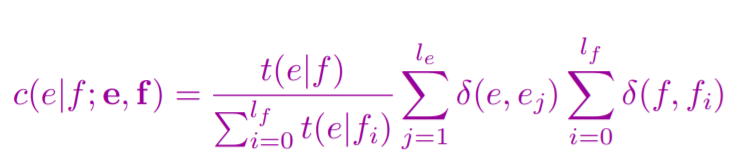
代码例子中我以F表示法文词，E表示英文词，因为这里alignment是一一对应的，不存在对应不上的NULL情况出现，因此我在例子中没有加入NULL position，但通常情况是要加的，请大家注意。这里通过uniform的方法初始化，语料比较小，我就写死了，通常情况是统计法文单词个数lf，初始化概率均为1/lf，以the英文单词为例，它对应所有法文单词的初始概率都为1/4，即25%，就是平均概率。代码中t表示transition probs参数，pairs是平行语料库，这里面是随便举的例子，不一定真实情况是这样，希望大家别钻牛角尖。

1. initialize c(e|f)，伪代码对应2-5行：

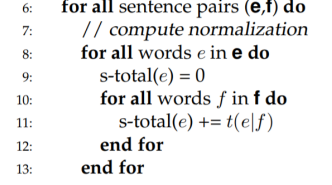


因为我们是通过统计平行语料库中，英文和法文对应的次数，来计算其对应的概率的，并且每一次循环都得初始化这个次数，重新统计，因此初始化c(e|f)的过程是放在循环中的，这里外层循环表示迭代计算次数，这里设置的20，total表示在语料中每个单词出现的期望次数（这里期望次数其实是计算出的期望值，在本例中，因为是通过计算出来的，所以因此可能是小数）sTotal表示句子中每个单词对应的期望次数，因此我们只有先计算sTotal期望，再计算整个语料库的total期望。

1. EM算法伪代码对应（6-28行）：

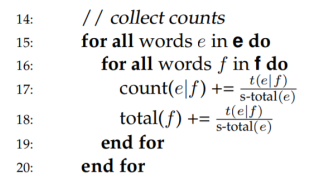
该步骤是实现EM算法迭代更新transition probs参数的方法，是通过这个式子进行计算的。因为我们要更新的参数是概率形式，在最后更新参数时需要做归一化操作

1. EM算法1（伪代码对应6-13行）：



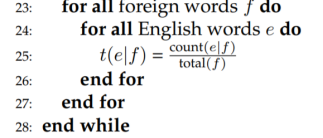
此步我们需要遍历平行语料库中的句子，将初始化的t(e|f)的赋值给初始化的sTotal概率。

1. EM算法2（伪代码对应14-20行）：



此步我们计算P(a|e,f)的期望，通过式子计算，注意，这里是针对句子计算，所以才是除以的是sTotal

(c) EM算法3（伪代码对应22-28行）：



这里就是归一化更新操作了，因为更新的是概率，我们统计出的是期望次数，因此要用sentence中的count除以总的count得到概率

这里以伪代码形式呈现是希望大家自己根据这个伪代码把代码填充上，这个伪代码说实话，已经是最后的代码了，大家补充一下就好，一定还是要过一下脑。时间比较赶，不知道有没有细节写错了，希望大家批评指正。