The basic idea of translation is word by word. Like

I   love you

我 爱  你

We only need probability of p(我 | I), p(爱 | love), p(你 | you), and the probability of “我爱你” as translated result is simply:

           p(我 | I) \* p(爱 | love) \* p(你 | you)

Or

          log(p(我 | I)) + log(p(爱 | love)) + log(p(你 | you))

and those p probabilities  can be derived from large corpus.

Besides this translation probability, we also have to consider language models, which is how likely "我爱你“ is a sensible Chinese sentence. as we learned in NLP(1), we can easily derive the language model from the large existing corpus. so the probability changes to

          log(p(我|\*,\*)) + log(p(爱 | \*,我)) + log(p(你 | 我，爱)) +

          log(log(p(我 | I)) + log(p(爱 | love)) + log(p(你 | you))

First line come from trigram language model, second line is from translation model.

This is the most straight forward way when think about translation. but sometimes different language have different words order, like

     Women who is beautiful

     美丽的女人

English often tend to put adjective behind noun, this kind of words reorder appears between any two languages, another example is Japanese always put verb at the end of sentence. So to take words reorder into account, linguist introduce a distortion term

q(i| j, l, m) – i : the ith word of result language (Chinese)

j : the jth word of source language (English)

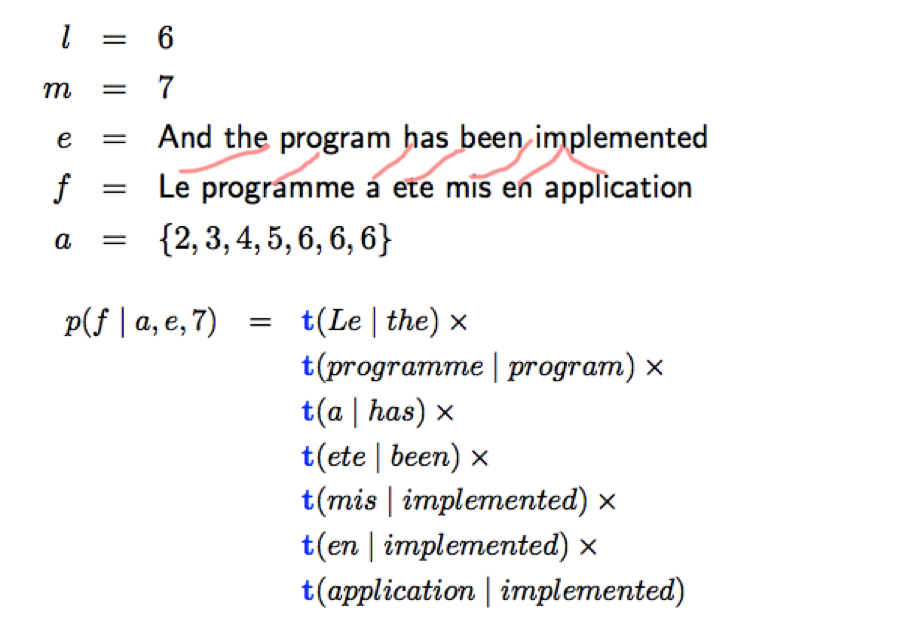
l : the length of result language sentence (Chinese)

m : the length of source language sentence (English)

This term represents the probability about alignment of word, then the probability become

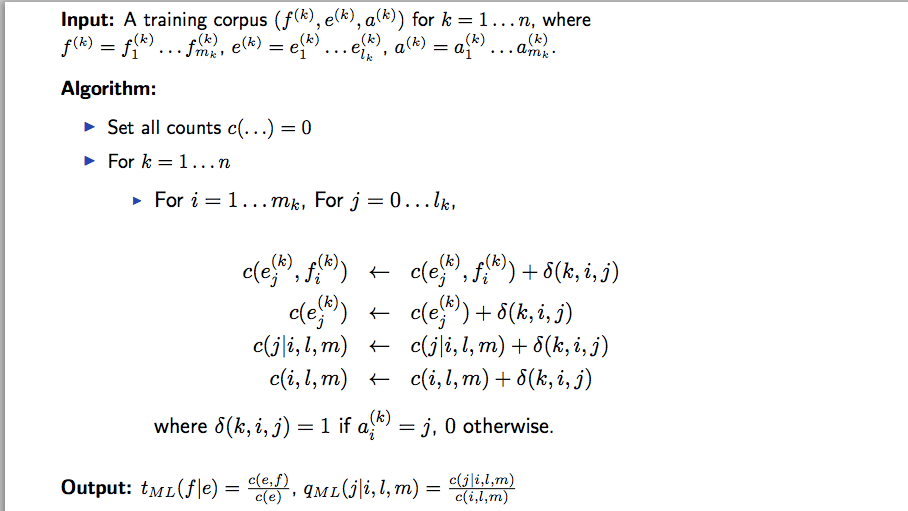
Distortion term (q) \* translation model (t) \* language model (p) (1)

Alignment means how words in source and target language are aligned, like the following picture:



As seen in equation (1), there are 3 parameters needed to evaluate the translation sentence, we already know how to train language model in NLP (1), then we have to train the distortion term (q) and translation model (t)

We only have k (k can be millions) bilingual sentence, c(k), e(k). What we want to obtain is q and t term. we can calculate q, t by following algorithm.

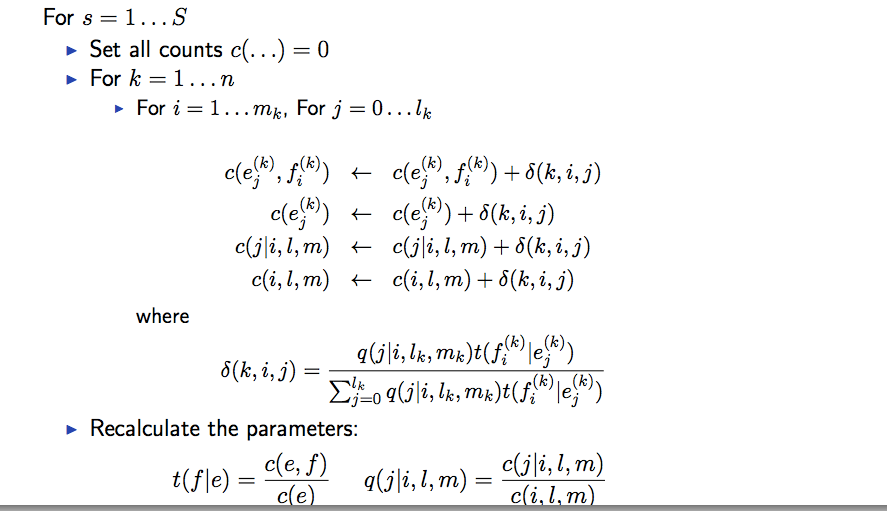


In algorithm above, we need matrix a, because delta(k, I, j) is directly related to a. but in the bilingual sentence, there is hardly exist such alignment matrix. In statistics, there is a method called EM algorithm, which can derive (or guess) hidden variable iteratively.

The procedure is like following:

1. Set q and t to some random value
2. Use q and t to derive alignment matrix a
3. Use matrix a to recalculate q and t
4. Repeat 1)-3) until q and t converge.

This algorithm ensure that q and t would achieve a local maximum of function F(q,t). Following are how to derive matrix a from q and t



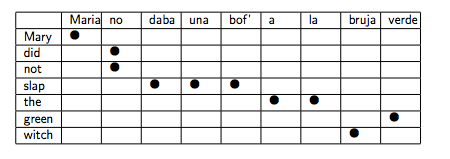
Actually we are recalculating delta coefficient, which indicates the probability of ith in target language word align with jth word in source language. The numerator is the likelihood of distortion term which j align with I, times the likelihood of word f(i) translated to word e(j), and then normalize this term by divide total likelihood.

Second part of machine translation is phrase based translation model.

Because in IBM model 2 (Above model which consisted of distortion term, translation term and language model term), the alignment only allow word in target language align to one word in source language that is a big deficiency. So linguist introduces phrase pair, which is many to many pair. Like:

“太棒了” – “So awesome”

To construct these phrase pair, we first need calculate the alignment matrix in both directions, which first use (Chinese, English) then use (English, Chinese) to get the alignment, then use a growing algorithm to obtain the alignment matrix, which is like following:



As we can see, each line and each column can have multiple dots that mean multiple pairs.

After obtain this alignment matrix, we can start translate sentence by these phrase.

Assume we have sentence

x1, x2, x3… xm

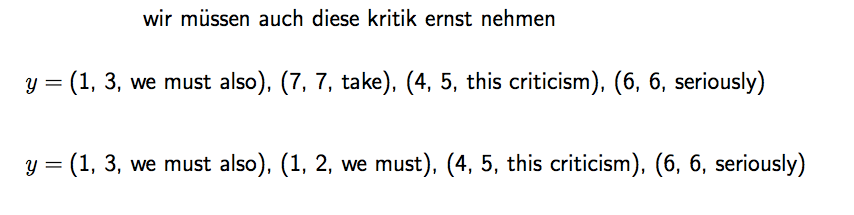
We need translate it to :

y1, y2, y3… yl

The score of result sentence is consisted of

Distortion term \* phrase pair likelihood \* language model

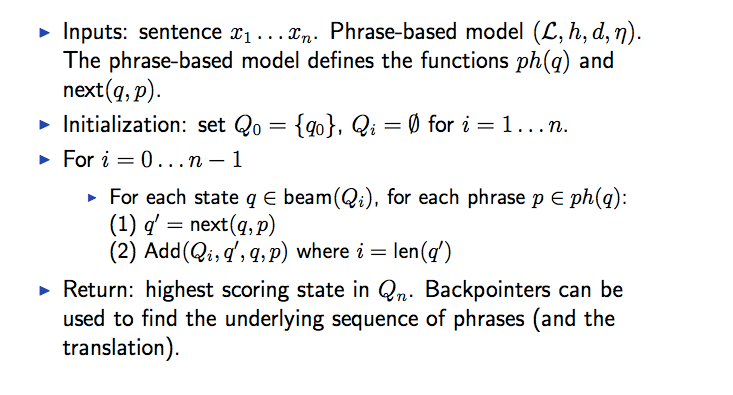
Which is generally same as IBM model with changes in translation model, and minor change in distortion term, now the result sentence is like following:



Each pair is (start, end, corresponding phrase), and would penalize when the end position of last pair is different with start position in current pair, like

(1, 3) (7,7) The penalization is -|3 + 1 – 7| = -3

Finally, we use a beam search algorithm to get the final result, because we may have lots of phrase pairs, we can only search some of combination. So we need strong cut in the process, beam just represents we only keep states whose score is no smaller than the highest one minus certain threshold, the rest states are abandoned. We also group states by the words they have aligned already, like following:



Qi represents states who has occupied i words in source language already.