

METL

TP3

Tientso Ning

IBM Model 1

1. Calculate the probabilities.

$$p(e, a|f) = \frac{1}{(l_f+1)^{l_e}} \prod_{i=1}^{l_e} t(e_i|f_{a_i})$$

$$p(e|f) = \frac{1}{(l_f+1)^{l_e}} \prod_{i=1}^{l_e} \max_j t(e_i|f_j)$$

$$\text{the house is small} = (0.7)(0.8)(0.8)(0.4)/(4+1)^4 = 0.00028672$$

$$\text{the house is little} = (0.7)(0.8)(0.8)(0.4)/(4+1)^4 = 0.00028672$$

$$\text{small house the is} = (0.7)(0.8)(0.8)(0.4)/(4+1)^4 = 0.00028672$$

$$\text{the} = 0.7/(4+1)^1 = 0.14$$

The IBM 1 model does not account for order of words (as we can see above, having the same probability for words out of order). Additionally, it does not penalize short translations accordingly, which we see above as well, where the short “the” receives a good score (the highest) compared to the translations before it.

The noisy channel model has a language model component, meaning it can penalize the grammatical aspects of the translation model using this language model concept. As seen before, it would penalize sentence 3, since the language model component will have sentence 3 rank lower due to grammatical incorrectness (compared to 1 or 2).

IBM Model 2

1. Calculate probabilities.

According to IBM Model 1, $p(e1, a1|g) = p(e2, a2|g)$ since IBM Model 1 doesn't take into account alignment, as we see before in the previous example. Under IBM Model 2, $p(e1, a1|g)$ should be more likely than $p(e2, a2|g)$ since we know that the first translation makes more sense to us, and this is due to the fact that IBM Model 2 takes into account alignment probabilities, and the alignment of the words are a bit more likely under alignment 1.

Provided calculations:

$$p(e1, a1|g) = p(m|l) * p(a1|j, l, m) * p(f|e)$$

$p(a1|j,l,m) = D(i=1, j=1, l=6, m=5) D(i=1, j=2, l=6, m=5) D(i=3, j=3, l=6, m=5) D(i=4, j=4, l=6, m=5) D(i=2, j=5, l=6, m=5) D(i=5, j=6, l=6, m=5)$

$p(f|e) = T(\text{natürlich}|of) T(\text{natürlich}|course) T(\text{das}|the) T(\text{Haus}|house) T(\text{ist}|is) T(\text{klein}|small)$

$p(e2, a2|g) = p(m|l) * p(a2|j, l, m) * p(f|e)$

$p(a1|j,l,m) = D(i=3, j=1, l=6, m=5) D(i=1, j=2, l=6, m=5) D(i=1, j=3, l=6, m=5) D(i=5, j=4, l=6, m=5) D(i=2, j=5, l=6, m=5) D(i=4, j=6, l=6, m=5)$

$p(f|e) = T(\text{natürlich}|of) T(\text{natürlich}|course) T(\text{das}|the) T(\text{Haus}|house) T(\text{ist}|is) T(\text{klein}|small)$

2. Most probable alignment under Model 2 can be computed efficiently if we follow fast_align:

- For each $i = 1, 2, \dots, l_e$ that is an empty word (dropped in translation), we consider it null with probability p_0
- If not, we use the log-linear distribution $\frac{e^{\lambda h(i,j)}}{Z_\lambda(i)}$ to choose a value for a_i
- For each $j = 1, 2, \dots, l_e$ choose a output word e_j according to $t(e_j|f_{a(j)})$

$$a(j|i) = \begin{cases} p_0, & j = 0 \\ (1 - p_0) \times \frac{e^{\lambda h(i,j)}}{Z_\lambda(i)}, & 0 < j \leq n \\ 0, & \text{otherwise} \end{cases}$$

$$h(i, j, m, n) = - \left| \frac{i}{m} - \frac{j}{n} \right|$$

$$Z_\lambda = \sum_{j=1}^n \exp(\lambda h(i, j))$$

We then are able to use these values in the E-step of the EM Algorithm

$$p(\mathbf{e}|\mathbf{f}) = \prod_{i=1}^m p(e_i|\mathbf{f}) = \prod_{i=1}^m \sum_{j=0}^n a(j|i) t(e_i|f_j), p(a|\mathbf{e}, \mathbf{f}) = \frac{p(a, \mathbf{e}, \mathbf{f})}{p(\mathbf{e}, \mathbf{f})}$$

The big-picture here is that originally, the Model 2 alignment probabilities are calculated based on parameters that are independent on each of the positions of the word in alignment, all conditioned on the lengths and positions, making it inefficient. The fast_align method instead calculates the parameters using a log-linear method.