

Lecture 6 — Jan 27, 2006

*Lecturer: Anoop Sarkar**Scribe: Maxim Roy*

6.1 Required Reading

1. Read section 34-37 of the workbook “A Statistical MT tutorial Workbook”, written by Kevin Knight.
2. Read the paper “A Systematic Comparison of Various Statistical Alignment Models”, written by Franz Josef Och and Hermann Ney.
3. Read the paper “The Mathematics of Statistical Machine Translation: Parameter Estimation”, written by Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra and Robert L. Mercer.

6.2 Road Map

We start with a Road Map of all the IBM models that we have covered so far and what we are going to cover.

Model 1: lexical translation (one parameter t)

Model 2: lexical translation + absolute reordering model

Model 3: adds fertility model

Model 4: relative reordering model

Model 5: fixes deficiency

HMM: HMM model where alignments are hidden

Model 6: Combines HMM + Model 5

The road map can also be view as below with respect to the models.

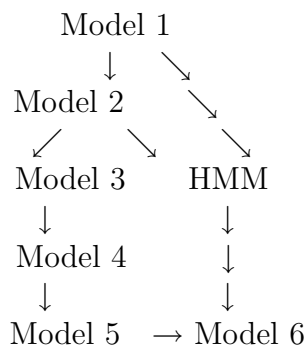


Figure 6.1. Road Map

After discussing all the IBM models, phrase based models and syntax based models will be discussed.

6.3 Generative picture

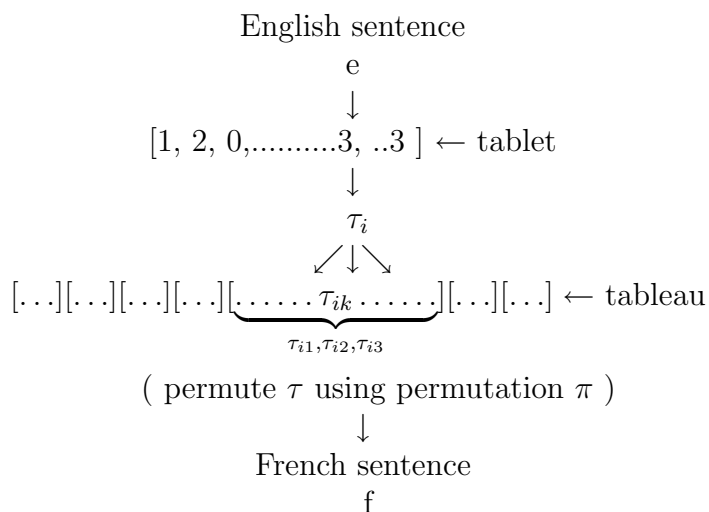


Figure 6.2. Generative picture

The joint likelihood for a tableau, τ , and a permutation, π , is

$$Pr(\tau, \pi|e) = \prod_{i=1}^l Pr(\phi_i|\phi_1^{i-1}, e)Pr(\phi_0|\phi_1^l, e) \times \prod_{i=0}^l \prod_{k=1}^{\phi_i} Pr(\tau_{ik}|\tau_{i1}^{k-1}, \tau_0^{i-1}, \phi_0^l, e) \times \prod_{i=1}^l \prod_{k=1}^{\phi_i} Pr(\pi_{ik}|\pi_{i1}^{k-1}, \pi_1^{i-1}, \tau_0^l, \phi_0^l, e) \times \prod_{k=1}^{\phi_0} Pr(\pi_{0k}|\pi_{01}^{k-1}, \pi_1^l, \tau_0^l, \phi_0^l, e) \quad (6.1)$$

All of the models are simplification of equation 6.1. Knowing τ and π determines a French string and an alignment, but in general several different pair τ, π may lead to the same pair f,a and the set of such pairs is denoted by (f,a), so

$$Pr(a, f|e) = \sum_{(\tau, \pi) \in (f, a)} Pr(\tau, \pi|e) \quad (6.2)$$

6.3.1 Model 4

Discussing Model 4 we will use the notation as used in Och and Ney paper (page-26). Let us consider the English sentence as $e_0, e_1, e_2, \dots, e_I$ and French sentence as $f_0, f_1, f_2, \dots, f_J$. For each i in English word let B be a set as $B: i \rightarrow B_i \subset 1, \dots, j, \dots, J$ which is also called tableau and $\Phi_i = |B_i|$ is the fertility of the word e_i . Now the distortion probability for Model 4 will be:

$$P(B_i|B_{i-1}, e_i) = P(\Phi_i|e_i) \cdot P_{=1}(B_{i1} - B_{p(i)}^- | \dots) \cdot \prod_{k=2}^{\Phi_i} P_{>1}(B_{ik} - B_{ik+1} | \dots) \quad (6.3)$$

Here $P_{=1}(\dots)$ is used to position the first word of a set B_i and $P_{>1}$ is used to position the remaining words from left to right. The function $i \rightarrow i' = p(i)$ gives the largest value $i' < i$ for which $|B_{i'}| > 0$ and the symbol $B_{p(i)}^-$ denotes the average of all elements in $B_{p(i)}$. In Model 4 every word is dependent on the previous aligned word and on the word classes of the surrounding words. One of the problems with Model 4 is that it may generate same French positions for different English words and also generate positions outside the length of the French word.

6.3.2 Model 5

Model 5 is discussed from the Brown et al. paper. It is almost the same as Model 4 except the last term $(1 - \delta(\nu_j, \nu_{j-1}))$ as described in the below 6.4 equation. Here ν_j is used to keep track of number of vacant position that exists. So in Model 5 we only map to vacant positions. Model 5 doesn't generate same French positions for different English positions like in Model 4. Therefore Model 5 takes care of the deficient problem.

$$\begin{aligned} Pr(\prod_{[i]k} &= j | \pi_{[i]1}^{k-1}, \pi_1^{[i]-1}, \tau_0^l, \phi_0^l, e) \\ &= d_{>1}(\nu_j - \nu_{\pi_{[i]k-1}} | B(f_j), \nu_m - \nu_{\pi_{[i]k-1}} - \phi_{[i]} + k)(1 - \delta(\nu_j, \nu_{j-1})). \end{aligned} \quad (6.4)$$



Definition of Deficiency: In statistical machine translation, when a model wastes some of its probability mass on impossible strings it is called deficient. A special problem of Model 3 and 4 concerns the deficiency of the model, which is, the same position can be chosen twice in the source string. Also a position before the first or beyond the last position may be chosen in Model 4. The deficiency of both models is removed in Model 5 by keeping track of vacant positions in the source string. We will again look at deficiency when discussing about parsing.

6.3.3 Model 2

Model 2 involved word translation like Model 1 but also uses distortion. Model 2 also finds Viterbi alignment efficiently as Model 1. But the distortion probabilities in Model 2 is different from Model 3. The distortion probability in Model 3 is $d(fpos|epos, l, m)$ where as in Model 2 the distortion probability is $d(epos|fpos, l, m)$. The $P(a, f|e)$ formula for Model 2 is :

$$P(a, f|e) = \prod_{j=1}^m t(f_j|eaj) \prod_{j=1}^m a(a_j|j, l, m) \quad (6.5)$$

Model 2 might penalize an alignment that connects positions that are very far apart. The algorithm to find the best Model 2 alignment is similar to Model 1 just multiplied the distortion factor.

Model 2 is more efficient and also produces better alignments than Model 1.

for $1 \leq j \leq m$

$$a_j = \underset{i}{\operatorname{argmax}} t(f_j|ei) * a(i|j, l, m)$$

6.3.4 Transformation of Parameters between Models

We can run Model 1 from several iterations and transfer values of t parameter as input to Model 2 instead of uniform t values. For first Model 2 iterations we will use uniform a values. While transferring parameter values from Model 2 to Model 3 we might like to use everything we learned in model 2 iteration to set the initial values for Model 3 parameters.

Now let us consider we want to compute initial fertility parameter values from Model 1 training. For example considering a corpus with the sentence pair bc/xy, suppose Model 1 has learned that $t(x|b) = 0.8$, $t(y|b) = 0.2$, $t(x|c) = 0.2$ and $t(y|c) = 0.8$. Now to guess the initial values for the fertility we look at four methods.

Method one just assigns uniform probability so $n(0|b) = n(1|b) = n(2|b) = 1/3$. Method 2 pretend that each alignment is equally likely and collects fractional counts of fertilities over all alignments so $n(0|b) = 1/4$, $n(1|b) = 1/2$ and $n(2|b) = 1/4$.

Method three collects fractional counts of fertilities over all alignments but take word-translation probabilities into account. So

$$P(\text{alignment1}|e, f) = 0.8 * 0.8 \Rightarrow \text{normalized} \Rightarrow 0.64$$

$$P(\text{alignment2}|e, f) = 0.2 * 0.8 \Rightarrow \text{normalized} \Rightarrow 0.16$$

$$P(\text{alignment3}|e, f) = 0.8 * 0.2 \Rightarrow \text{normalized} \Rightarrow 0.16$$

$$P(\text{alignment4}|e, f) = 0.2 * 0.2 \Rightarrow \text{normalized} \Rightarrow 0.04$$

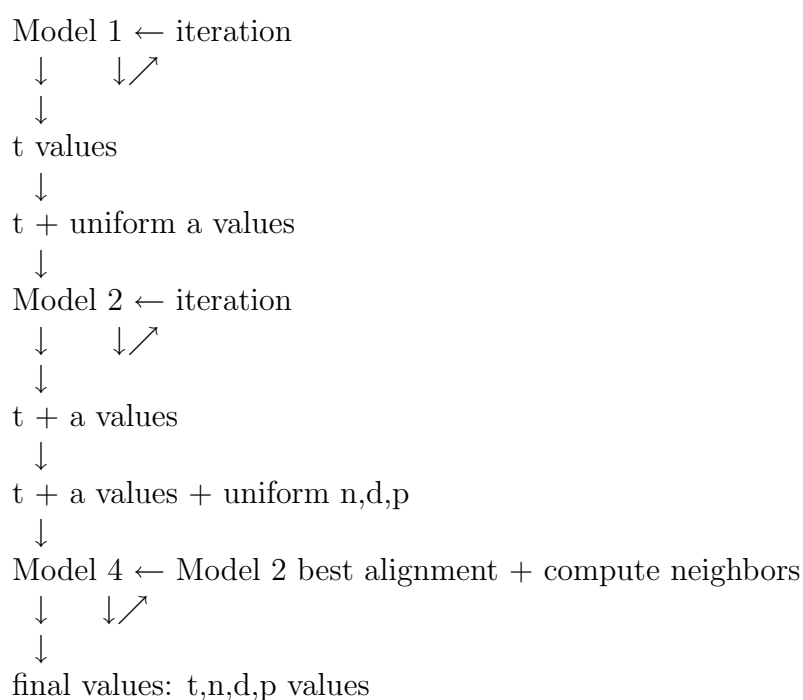
Now the fractional counts are $nc(0|b) = 0.16$, $nc(1|b) = 0.64 + 0.04 = 0.68$ and $nc(2|b) = 0.16$. As the values of nc count happen to already add to one so they remain the same after normalization: $n(0|b) = 0.16$, $n(1|b) = 0.68$ and $n(2|b) = 0.16$. In Method 3 distribution is better than method 1 and 2 as it takes into account what Model 1 has learned.

As method 3 does not scale to long sentences we use Method 4 which gives

us exactly the same results as Method 3 with a computationally cheaper algorithm. The algorithm is described in section 35 of Kevin Knight workbook. The proof of Method four can be found after (Brown et al, 1993)'s equation 108.

6.4 Final training Scheme

So now our final training schema looks like this:



References

- [1] P. E. Brown , S. A. Della Pietra, V. J. Della Pietra and R. L. Mercer,1993. The Mathematics of Statistical Machine Translation. Computational Linguistics, 19(2):263–311.
- [2] K. Knight. 1999. A Statistical MT tutorial Workbook. JHU CLSP summer workshop.

- [3] F. J. Och, H. Ney. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics* 29(1): 19-51.