Programming Problems (due November 15th at 5pm)

Please read the submission instructions, policy and hints on the course webpage.

In this programming problem you will implement IBM translation models 1 and 2 and use your implementation to learn word alignments in an English/German parallel corpus.

The two files *corpus.en.gz* and *corpus.de.gz* contain 20,000 English and German sentences respectively. The i-th sentence in the English file is a translation of the i-th sentence in the German file. The files are in one-sentence-per-line format (but compressed using *gzip*). Words in each line are separated by single spaces.

Question 4 (25 points) - IBM Model 1

Recall that IBM model 1 only has word translation parameters $t(f|e)$, which can be interpreted as the conditional probability of generating a foreign word f from an English word e (or from NULL).

We can estimate $t(f|e)$ using the EM algorithm (see handout).

Implement a version of IBM model 1, which takes *corpus.en.gz* and *corpus.de.gz* as input.

- Your implementation should only store t parameters for possible pairs of foreign and English words (i.e. words that occur together in a parallel translation) and the special English word NULL.

  In the initialization step set $t(f|e)$ to be the uniform distribution over all foreign words that could be aligned to e in the corpus.

  More specifically

  $$t(f|e) = \frac{1}{n(e)},$$

  where $n(e)$ is the number of different words that occur in any translation of a sentence containing e. Note that the special English word NULL can be aligned to any foreign word in the corpus.

- Starting from the initial $t(f|e)$ parameters, run 5 iterations of the EM algorithm for IBM model 1 (this may take a while). Then, for each English word e in *devwords.txt*, print the list of the 10 foreign words with the highest $t(f|e)$ parameter (and the parameter itself). Submit your code and the result.

- Finally, use your model to find alignments for the first 20 sentence pairs in the training data. For each sentence, align each foreign word f_i to the English word with the highest $t(f|e)$ score, i.e.

  $$a_i = \arg \max_{j \in \{0 \cdots l\}} t(f_i|e_j).$$

  Print the alignments as a list of m integers containing the a_i values.

Question 5 (25 pts) - IBM Model 2

We will now extend our alignment model to IBM model 2 by adding alignment parameters $q(j|i, l, m)$.

- Initialize the q parameters to the uniform distribution over all j for each i, l, and m, i.e.

$$q(j|i,l,m) = \frac{1}{l+1}$$

You only need to store parameters for pairs of sentence lengths l and m that occur in the corpus. To initialize the $t(f|e)$ parameters, use the last set of parameters (after 5 iterations) produced by your implementation of IBM model 1.

- Extend your implementation of the EM algorithm to IBM model 2. Adapt the delta function to include $q(j|i,l,m)$ parameters

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

and compute expected counts $c(j|i,l,m)$ and $c(i,l,m)$.

After each iteration through the corpus re-estimate the $t(f_i, e_j)$ parameters as before and the q parameters:

$$q(j|i,l,m) = \frac{c(j|i,l,m)}{c(i,l,m)}.$$

Then, run 5 iterations of EM for IBM model 2.

- As before, use the model to compute alignments for the sentence pairs in the corpus. For each foreign word f_i, the best alignment is

$$a_i = \arg\max_{j \in \{0 \cdots l\}} q(j|i,l,m)t(f_i|e_j).$$

Print the alignments for the first 20 sentence pairs as before and comment on the difference in model performance. Submit your code.

Question 6 (25 pts) - Finding translations

Claudia Clumsy dropped her German/English parallel transcripts of a European Parliament debate, so that the sentences are no longer aligned. English sentences are in the file *scrambled.en*, German sentences are in *original.de*. Use your trained IBM Model 2 from the last question to recover the original English sentence by computing

$$\arg\max_e \max_a P(f,a|e)$$

for each German sentence, where

$$P(f,a|e) = \prod_{i=1}^{m} q(a_i|i,l,m)t(f_i|e_{a_i})$$

That is for each German sentence find the English sentence that produces the highest-scoring alignment.

Note: You may run into underflow issues when aligning. To avoid this problem we recommend using log-probs, i.e. solve the following problem

$$\arg\max_e \max_a P(f, a|e) = \arg\max_e \max_a \log P(f, a|e) = \arg\max_e \max_a \sum_{i=1}^{m} \log(q(a_i|i, l, m)t(f_i|e_{a_i}))$$

When the inner product $q(a_i|i, l, m)t(f_i|e_{a_i})$ is zero, you can substitute a large negative constant.

- Print out a new file `unscrambled.en` with the best sentence match from `scrambled.en` in the order of the original German sentences. You should run `python eval_scramble.py unscrambled.en original.en` to check the accuracy of your solution. Comment on the efficiency of your method and any issues you saw while unscrambling.