Run commands, notes and motivations

Run commands

Below python script will run for about 80 minutes, during which consume 7.5G memory at most.

```
python answer/align.py -p europarl -f de -n 100000 > output.a
head -1000 output.a > upload.a
```

Notes and motivations

The core solution to this language alignment is to use training data to find a most likely from `f_i` to `e_j` .

- Core is EM algorithm. Stage expectation collects counts based on already known and training data, then Maximization stage use viterbi alg. to find the most possible pair.
  - Expectition: $Pr(f|e,t) = \prod_{i=1}^{I} \sum_{j=1}^{J} t(f_i|e_j)$
  - Maximization: $argmax_t L(t) = argmax_t \sum_s logPr(f^{(s)}|e^{(s)},t)$
- Baseline implements ibm model 1, trains the model to generate parameters $t(f_i|e_j)$ for each word pair, and uses viterbi algorithm to find the most possible word pair.

$$t_k(f_i|e_j) = \sum_{s=1}^{N} \sum_{(f_i,e_j)\in(\mathbf{f}^{(s)},\mathbf{e}^{(s)})} \frac{count(f_i,e_j,\mathbf{f}^{(s)},\mathbf{e}^{(s)})}{count(e_j,\mathbf{f}^{(s)},\mathbf{e}^{(s)})}$$

$$count(f_i,e_j,\mathbf{f},\mathbf{e}) = \frac{t_{k-1}(f_i|e_j)}{\sum_{a_i=1}^{J} t_{k-1}(f_i|e_{a_i})}$$

$$count(e_j,\mathbf{f},\mathbf{e}) = \sum_{i=1}^{I} count(f_i,e_j,\mathbf{f},\mathbf{e})$$

- First improvement is to train two models French->Enghlish and English->French, and then generate optimal parameters from the intersection of two models.
  $opt_{res} = params(e|f) \bigcap params(f|e)$
- Modify the baseline from 'go over each word in a sentence' to 'go over the word set of this sentence'. The intuition behind this is simple, IBM model 1 only considers word pair, and pays no attention to the position. Thus, set will quicken training descently.
- Diagonal alignment TODO: @SHUO
- IBM model 2
  - Given a French sentence $\mathbf{f} = f_1...f_n$ and English sentence $\mathbf{e} = e_1...e_m$, we model alignments of the form $\mathbf{a} = a_1...a_n$, where each a_i takes a value from 1 to m, denoting the index of the English word to which the ith French word is aligned. Lexical translation parameters t and this a work together to provide a possibility for each word pair.

$$p(\mathbf{f},\mathbf{a}|\mathbf{e}) = \prod_{i=1}^{n} p(a_i = j|i,j,m,n) \times p(f_i|e_{a_i})$$

$$p(a_i = j|i,j,m,n) = \frac{1}{Z_{i,m,n}} e^{\lambda h(i,j,m,n)}$$

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

  - Use IBM model 1 output as model 2 input initilization.
- Smoothing TODO: @Shanghao