Sarika Halarnakar - shalarn1
Sindhuula Selvaraju - sselvar4

Machine Translation Assignment 2: Word Alignment
2.25.16

Part 1: IBM Model 1

Usage: (In Aligner)
python align_IBM -n 1000 > ibm.a
python score-alignments < ibm.a

Part 2: HMM + some experimentation

Usage: (In Aligner)
python align_HMM -n 1000 > hmm.a
python score-alignments < hmm.a

Mathematical Desciption:

We decided to implement the HMM Alignment Model(It is our version of created by tweaking the IBM Model 1 that was designed). In an HMM Alignment, we assume that there is typically a strong localization effect in aligning the words in parallel text. That is, alignments have a strong tendency to remain in the local neighborhood of its source when translated into another language.

To implement an HMM model, we introduced a mapping of j -> i which assigns a French word in position j to an English word in position i.

The mathematical model captures the strong dependence of i on the previous alignment, thus the probability of alignment i for position j has a dependence on the previous alignment i' and the length of the English sentence:

$$p(i|i', I) = \frac{s(i - i')}{\sum_{l=1}^{I} s(l - i')}$$

where l is the length of the sentence in English and s(i-i') is the set of non-negative parameters.
From this we get that the calculation of the posterior probability is now

$$Pr(f_1^J|e_1^I) = \sum_{a_1^J} \prod_{j=1}^{J} \left[p(a_j|a_{j-1}, I) \cdot p(f_j|e_{a_j})\right]$$

where $i = a_j$ and $i' = a_{j-1}$

Taking this equation, we simply replaced the equivalent equation from our IBM Model 1. We achieved the following statistics:

Using IBM Model 1:
Running for 100 Lines:
Precision = 0.553279
Recall = 0.245562
AER = 0.625430

Running for 1000 Lines:
Precision = 0.598039
Recall = 0.337278
AER = 0.538820

As can be seen the AER vastly improves from 100 lines to 1000 lines.

Using HMM Model:

Though the HMM model is not better than the IBM model this may be due to the fact the the jump_key being calculated is incorrect with respect to the language i.e. the words may have been reordered differently than calculated. Also, since we experimented with our attempt at creating an HMM model by changing different parameters the best one suited to our data may not have been found by us.

Running for 100 Lines:
Precision = 0.521401
Recall = 0.198225
AER = 0.662185

Running for 1000 Lines:
Precision = 0.482639
Recall = 0.195266
AER = 0.672524

Runnning for 2000 Lines:
Precision = 0.506944
Recall = 0.198225
AER = 0.659744

As seen above, our alignment model doesn't work better than the IBM Model 1. Also as observed, the AER for 100,1000 and 2000 lines is almost the same so for irrespective of the size of training set it still gives the same AER.

NOTE: We were not able to run for higher number of lines because it took too long resulting in either out of memory error or the process being automatically killed.