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

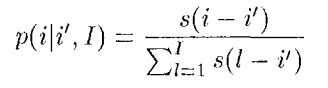
python score-alignments < hmm.a

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.

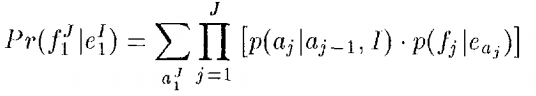
To implement an HMM model, we introduced a mapping of j -> i which assigns a French word in position j to an English word in position i.

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:



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



where i = aj and i’ = aj-1

Taking this equation, we simply replaced the equivalent equation from our IBM Model 1.

We achieved the following statistics for 1000 sentence alignments:

Using IBM Model 1:

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.