The WordAlign constructor first reads in the file, splitting the sentences into words, runs iterations, and then prints the outputs. The file reading and output printing clearly works, as my program works as expected on the example from class (with “la casa” and “casa verde”). Thus the only method we need to analyze is the runIteration() method. We will first analyze the code, and then verify that the code works on examples.

1. Code Analysis

This method only contains approximately 30 lines of code, so we can step through it to ensure everything looks good. For each sentence, we loop through all foreign words. You can imagine the following layout for the sentence:

e1 e2 e3
f1 f2 f3 f4 f5

For the sake of efficiency, we calculate the probability of any english word getting matched to the foreign word outside of the loop through the english words. Then, for each english word, the probability of that english word getting matched to the current foreign word is just p(e,f) divided by the pre-computed denominator. We increment the corresponding partial counts.

So far, this logic corresponds precisely to the efficient EM algorithm discussed in class. Our last step is to recompute the probabilities and normalize.

2. Examples

The code works both on the example given in class (with “la casa” and “casa verde”), and also produces mostly correct translations on a corpus of 10k sentences.

I also ran the following example using French sentences:

English	French
I come from Boston	je viens du Boston
I come home	je viens a la maison
the big home	la grande maison
the big woman	la grande femme
I eat the chicken	je mange la poule
you eat the pie	tu mange la tarte
you are the big woman	tu es la grande femme
you come from China	tu viens du Chine

The output after 100 iterations was as follows:

big	grande	0.9999992258863848
chicken	poule	0.9822862123508006
woman	femme	0.9992907356794994
I	je	0.9921900153779198
come	viens	0.9828207560228852
pie	tarte	0.9855771681664753
home	a	0.31627085847536696
home	maison	0.6837291415242259
the	la	0.8758716243132079
the	grande	0.1241282390846463
are	es	0.9847550854636287
China	Chine	0.9741508608813926

eat	mange	0.9999999999992994
from	du	0.9705791405720892
you	tu	1.0
NULL	la	0.9999842247667327
Boston	Boston	0.9729356362379936

The translations are all correct. For english words with multiple options above the 0.1 threshold, the maximum likelihood option is always correct.