Code Documentation:

To use the program just use the command line as instructed in the writeup. To change the max depth of the decision trees, the number of stumps, and size of the stumps in the adaboost ensemble simply change the constants found at the top of the program labeled MAX_TREE_DEPTH, NUM_STUMPS, STUMP_DEPTH, respectively.

Features:

To choose the 19 features, I looked up the most common 3-5 letters from each language. I chose 3+ as 1-2 letters can be found coincidentally in various words from both languages. The features are: the, for, his, that, ther, here, with, from, this, able, aa, het, tot, van, dat, niet, zijn, heeft, and voor. I tried to get a proper variety of both English and Dutch words. More words could have been found, but it seemed a bit overkill.

Decision Tree Learning:

The decision tree loops through each feature, splitting the example data set off of whether the example has the feature or not, the entropy is then calculated and when the loop is done, the feature index with the best entropy is added to the tree and the function is recursive for each child until the max depth is reached, or all of the features have been tested. For the max depth I kept running the program with different depths until the error percent began to decrease. The depth I ended with was 19 for all 19 features as with all of the features it gets the best probability.

Boosting:

For ada boost, I initiate all of the examples with a weight of 1/len(examples). Then loop through and find weighted hypothesis' n times, where n is the set number of stumps, from the top of the file. For every loop, it finds a weighted decision tree and finds the error. From that it goes through and changes the weight of each example according to the error, and then normalizes the weights and adds the weighted hypothesis to a list. After testing, I found 10 stumps with a depth of 1 was enough.