Report for Assignment1:

1. Charniak suggests that utterance length may help distinguish the translations from the original utterances. For what proportion of utterance pairs in the *good-bad* file is the original shorter (fewer tokens) than the translation?

   71.83 % of the original utterances are shorter than the translation or 83 % is shorter or equal to the translation.

2. Create a smoothed unigram language model with the smoothing parameter alpha = 1. Compute the log probability of our language-model test data.

   ```
   -243967.47
   ```

3. Now set the unigram smoothing parameter alpha to optimize the likelihood of the held-out data as described in the text. What values of alpha do you find? Repeat the evaluation described in the previous step using your new unigram models. The log probability of the language-specific test data should increase.

   Started with alpha = 0.010 and incremented by small values (0.010). For alpha = 9.23 the model attained the maximum likelyhood. (Supporting data in the log.txt file)

4. Now try distinguishing good from bad English. In good-bad.txt we have pairs of sentences, first the good one, then the bad. Each sentence is on its own line, and each pair is separated from the next with a blank line. Try guessing which is which using the language model. This should not work very well.

   From the accuracy I'm getting from both the unigram and the bigram model, the accuracy of the bigram model is greater than the unigram. Hence the good translation may be from the bigram model(since the overall number of good translation sentences has higher probability in the bigram model).

5. Now construct smoothed bigram models as described in the text, setting beta = 1, and repeat the evaluations, first on the testing text to see what log probability you get, and then on the good and bad sentences.

   For beta = 1, for test set "TOTAL LOGLIKELYHOOD = -4796389.5 (log base 10)" and then for the whole good and bad sentences data total loglikelyhood is   -9116100.0

6. Set the bigram smoothing parameter beta as described in the text, and repeat both evaluations. What values of beta maximize the likelihood of the held-out data?

   For the beta values of around 50, the loglikelihoods are getting saturated.

7. Lastly, use the smoothed bigram model to determine good/bad sentences.

   My Bigram model’s accuracy is 74.9%

8. Print out a few utterance pairs your system gets wrong. What sorts of information would help you get these right?

```
Tuesday , February 10 , 1997
the tuesday 10 february 1998


senators ' STATEMENTS
' of senators

opening of 1998 Games at Nagano , Japan
the openness of the games of 1998 to nagano , to the japan
```

From the observation of the above sentences, treating the numbers and new words as unknown words may help in improving the accuracy.

The overall results of the system have been answered in the QA section. Implemented everything in Java as I’m new to Python . As described in class, the system’s performance can be improved by substituting the numbers with the NUM tag (which include date and time fields). Otherwise each unique number will be considered as a unique token and will result in data sparsity issue. But the domain should also be considered before making any such decisions.