1.1

$argmax_{s\in S}P(POS_{i-2} = VERB|s) = s_1$ Since:
For s_1: $P(POS_{i-2} = VERB|s_1) = \frac{2}{2} = 1$
For s_2: $P(POS_{i-2} = VERB|s_2) = \frac{1}{3}$
For s_3: $P(POS_{i-2} = VERB|s_3) = 0$

$argmax_{s\in S}P(POS_{i-2} = NOUN|s) = s_2$ Since:
For s_1: $P(POS_{i-2} = NOUN|s_1) = 0$
For s_2: $P(POS_{i-2} = NOUN|s_2) = \frac{2}{3}$
For s_3: $P(POS_{i-2} = NOUN|s_3) = 0$

$argmax_{s\in S}P(POS_{i-2} = NUM|s) = s_3$ Since:
For s_1: $P(POS_{i-2} = NOUN|s_1) = 0$
For s_2: $P(POS_{i-2} = NOUN|s_2) = 0$
For s_3: $P(POS_{i-2} = NOUN|s_3) = 1$

1.2

$abs(log\frac{Sense1|POS_{i+1}=NOUN}{Sense2|POS_{i+1}=NOUN}) = abs(log\frac{2}{3}) = 0.1791$

$abs(log\frac{Sense1|w_{i+1}=art}{Sense2|w_{i+1}=art}) = abs(log\frac{1}{2}) = 0.301$

2.2

Overview: The code was split into four parts: parsing, attaining the features, training, and evaluating. I used nested lists to hold the tags and words for each data point. I then acquire the bags of words and parts of speech by choosing the ones that appear more than threshold times, which will be later used for the co-occurrence models. In the models presented, I chose a threshold of 10 for the bag of words and a threshold of 20 for the parts of speech. This is because unlike words, each of the parts of speech appears more frequently, and a lower threshold will result in the inclusion of nearly all POS elements. After setting up, I acquire the individual features by selecting windows. The window size I selected was 4. In my opinion, this window size, while avoiding conflict with the relatively high threshold for bag of words, is also sufficiently small such that it wouldn't cause the problem of overfitting. I then feed the features into the naive bayes model and classify the test set with the trained classifier.

After training all six models, I got resulting accuracies that are roughly between 30 and 40 percent. In particular, the model trained with both the collocation and the cooccurrence features with words only performs the best(with a f-1 score of 0.451 and an accuracy of 0.379). The improvement achieved by using both of the feature vectors corresponds to my hypothesis that a combination of collocation and cooccurence will provide a more detailed, accurate, and holistic feature for word sense disambiguation. In addition, from these models, we can see that the word models generally do better than the ones with both. I think this is because of the fact that POS tagging is really general, thus not really contributing to training the model and improving the accuracy. Another problem that I think I might have run into is overfitting. I attempted to run the classification with the training set, and attained accuracy that's nearly 100 percent. Such high percentage could be a sign of overfitting. Thus, we need to think of ways to generalize more.

```
For collocation with words:
accuracy: 0.367816091954
precision: 0.414792008758
recall: 0.211664443076
f-1 score: 0.276035395001

For collocation with words and pos:
accuracy: 0.247311827957
precision: 0.187275985663
recall: 0.102976190476
f-1 score: 0.112905452036

For coccurence with words:
accuracy: 0.310344827586
precision: 0.321428571429
recall: 0.120381773399
f-1 score: 0.263392857143

For coccurence with words and pos:
accuracy: 0.310344827586
precision: 0.28549382716
recall: 0.120381773399
f-1 score: 0.268835140395

For both with words:
accuracy: 0.379310344828
precision: 0.400327868852
recall: 0.145628078818
f-1 score: 0.45089471494

For both with words and pos:
accuracy: 0.310344827586
precision: 0.32824933687
recall: 0.119150246305
f-1 score: 0.372938689218
```