Problem 1: Pruning Language Models

For this exercise, we will train a language model using absolute discount smoothing and pruning. You can complete this task by extending the code from your solution to Part 2 of Exercise 5.
+ Implement a method prune in the discounting model class, which takes epsilon as an arugment, and creates and returns a new absolute discounting model with only the bigrams and unigrams with a probability of at least epsilon in the original model.
+ Using the text /twain/pg119.txt for training, and /twain/pg3176.txt for testing, compute the perplexity for the original absolute discounting model, and the pruned models for epsilon = 10^{-n}, for n = 3, 4, 5, 6. Use d = 0.9. Report both the perplexity and the number of parameters used for each model.

We got the following result:
The perplexity for the original absolute discounting model: 1132.83561733

The perplexity for the pruned model with EPSILON = 0.001 : 7803.9018789
The perplexity for the pruned model with EPSILON = 0.0001 : 7937.37399109
The perplexity for the pruned model with EPSILON = 0.00001 : 2232.46314475
The perplexity for the pruned model with EPSILON = 0.000001 : 1132.83561733

Instruction to run the the code (for Linux):

```
$ python prb1.py
```

Problem 2: Classification

In this task, you will train a simple Naive Bayes classifier to perform author identification, using texts by Mark Twain and Jane Austen.
+ Using the provided materials in /twain/ and /austen/, create a Naive Bayes classifier using the word frequencies as features. For the class probabilities, you may assume the document counts are representative. Classify the excerpts in the /test/ folder by author, and report the results.

We got the following result:
Class for ./ex-6-materials/test/test1.txt : Mark Twain
Class for ./ex-6-materials/test/test2.txt : Jane Austen
Class for ./ex-6-materials/test/test3.txt : Mark Twain

+ What do you think of using the unigram probabilities as features for classification? Give an example (not necessarily from the given texts) of a case where unigram probabilities would not produce optimal results for classification.

Unigram probabilities as feature can be used for classification because its computational advantage. But other more advanced n-gram models outperform in accuracy. The reason for it is that it does not take into account the context of the term. Example of part speech tagging where each tag has a strong dependence of previous one.

+ Name a drawback of the Naive Bayes classifier in general, and explain at least one reason why, despite this, it is a useful model in practice.
One of the drawbacks of Naive Bayes classifier is conditional independence assumption. The model makes a strong assumption that features are independent of one another given some class, but in fact, they are not independent. The formula of calculating the probability belonging of X with f_1, f_2 and f_3 features to c class is calculated by multiplying probability of each feature as independent from other.

$$p(f_1 f_2 f_3) = \prod_{i=1} p(f_i|c) \tag{1}$$

Instruction to run the the code (for Linux):

```
$ python prb2.py
```