1 Solution

1. Unigram:
   (a) stored data type: dict[term:frequency]
   (b) choose the candidate with higher frequency
2. Bigram:
   (a) stored data type: dict[(word1,word2):frequency]
   (b) choose the candidate with higher frequency, however there will be zero possibility because the words may not been seen.
   (c) only count the back off of the one word before and after candidate because other words in the sentence share the same weight and can be treated as constant.
3. Bigram - add 1 smooth:
   (a) stored data type: dict[(word1,word2):frequency]
   (b) choose the candidate with higher frequency
   (c) there will not be any 0 possibilities because we assume every word had 1 frequency.

2 Result

1. Unigram:
   (a) Correctness: 0.6
   (b) it is just guessing with the 1/2 chance to get the correct one that is more frequent.
2. Bigram:
   (a) Correctness: 0.8
   (b) use the back off method to count the more precise possibility. As expected,there are 2 word with equivalent 0 possibility.
3. Bigram - add 1 smooth:
   (a) Correctness: 0.9
   (b) after applying the add 1 smoothing method, the possibilities become more accurate,and there is no 0 possibility.
   (c) there is still one sentence with the wrong answer might because we assume that the word is only dependent on the word before it. However the other word in the sentence might also influence the meaning of the sentence so that the different choice will be made.