Assignment 3

We test our function for several cases:

1. First, with deterministic as True we see test the case provided to us. The list ['she', 'was', 'not'] with a trigram model is successfully completed to becomes ['she', 'was', 'not', 'in', 'the', 'world', '.'].
2. The same function with a stochastic Markov model of 4-grams returns a mu ch more meaningful sentence. The same sentence becomes
   [‘she’, ‘was’, ‘not’, ‘in’, ’the’, ‘habit’, ‘of’, ‘seeing’, ‘much’, ‘occupation’, ‘at’, ‘home’, ‘,’, ‘and’ ‘you’]
3. To test whether our function executes stupid back off method successfully we initiate a tokenize sentence with a random name [‘Anas’, ‘natural’]. With a trigram model this function returns:
   ['Anas', 'naturally', 'attended','these','parties','were','exactly','calculated','to ','give','her',',',’and’'the','two']

Now we test the function with deterministic as False. The following cases:

1. ['she', 'was', 'not'] becomes ['she','was','not','were','the','was','does','by',
2. 'temper','advice',',','his','he','it','and']
3. [‘Anas’, ’naturally’, ‘becomes’] becomes ['Anas', 'naturally', 'becomes', 'out', 'minute','lady','here','the','your','dashwood','two','great','they','marriage', 'giggling']

NOTE: With deterministic being True, the initial tokenized list should have at least n-1 elements. The reason is that the function is designed as such that to make n-grams, it needs to look at n-1 words behind. Therefore, if the initial tokenized list has less than n-1 elements, the function wont work. One way to fix this is to add ‘<start>’ element appropriate number of times before the first elem ents of the list.