IDS 703
HW Assignment III
Submitted by Jiahui (Heather) Qiu
September 28, 2022

My Python script consists of three functions: (1) The get_ngram_counts function goes through all n-gram tokens in the corpus and counts the number of times each token appears. (2) The get_next_word function calculates the relative frequency of a word appearing after a prefix. If a higher-order n-gram has a zero count, the function will then back off to a lower-order n-gram and repeat the same process. If the deterministic flag is true, the function returns the word with the highest probability of appearing after a prefix. Otherwise (i.e., stochastic mode), it returns a randomly chosen word from all possible options with a probability of greater than zero. (3) The finish_sentence function calls the get_next_word() function and returns an extended sentence until the first period, question mark, or exclamation mark is found, or the sentence reaches ten total tokens.

Using the provided corpus (i.e., austen-sense.txt), I applied my text generator function to the following text cases, and the results are listed below:

Index	Sentence	N	Deterministic	Result
1	["she", "was", "not"]	3	True	['she', 'was', 'not', 'in', 'the', 'world', ',', 'and', 'the', 'two'] OR ['she', 'was', 'not', 'in', 'the', 'world', '.']
2	["she", "was", "not"]	3	False	['she', 'was', 'not', 'advancement', 'willing', 'obey', 'summoned', 'released', 'convert', 'rank'] and many other results
3	['I', 'like', 'fried', 'chicken']	4	True	['I', 'like', 'fried', 'chicken', 'out', 'of', 'doors', ';', 'and', 'marianne'] and many other results
4	["The", "restaurant", "is", "open", "on"]	5	False	['The', 'restaurant', 'is', 'open', 'on', 'thinks', 'had', 'held', 'summoned', 'unpardoned'] and many other results

The first test case returned two distinct sentences at different lengths when run many times. This is because a period and a comma have an equal likelihood of appearing after ['the', 'world']. The sentences have two different lengths due to the stopping criteria in my third function: the first sentence returned when its length reached ten tokens, while the second sentence stopped incrementing when it hit a period. You will observe the same behavior in other test cases for similar reasons.

The sentences returned from the first test case arguably make sense in English. However, results from the remaining test cases are far less ideal. Comparing the first two test cases, we can see that there are indeed values to generate words with the highest probabilities of occurrence after a prefix (i.e., when Deterministic = True). This approach, however, is not bulletproof as shown in the third test case. The difference in the text generator performance is primarily due to our training corpus and the coding logic. The results will likely improve with a larger corpus and by using smoothing techniques, rather than stupid backoff when addressing the problem of zero frequency n-grams.