NLP Assignment 1 – Tokenization & Counting

1. For each corpus (i.e., gutenberg.txt.utf8.gz and reuters.train.txt.gz), how many types, tokens, and hapax legomena (types that occur once) are in the corpus?
    a. gutenberg.txt.utf8.gz
        i. Types: 54851
        ii. Tokens: 2558994
        iii. Hapax legomena: 25658
    b. reuters.train.txt.gz
        i. Types: 30349
        ii. Tokens: 1216561
        iii. Hapax legomena: 11727
2. For each corpus, what are the 20 most frequent types?
    a. gutenberg.txt.utf8.gz
        i. , 187045
        ii. the 133170
        iii. and 94722
        iv. . 75140
        v. of 71158
        vi. to 47570
        vii. : 47563
        viii. a 33561
        ix. in 33385
        x. i 28458
        xi. that 28219
        xii. ; 27718
        xiii. he 25404
        xiv. it 21563
        xv. his 21370
        xvi. for 19395
        xvii. was 18646
        xviii. with 17554
        xix. not 17263
        xx. " 16345
    b. reuters.train.txt.gz
        i. . 67236
        ii. , 52338
        iii. the 51384
        iv. to 27310
        v. of 27306
        vi. in 22003
        vii. and 18968
        viii. said 18844
        ix. a 18631
        x. mln 13089
        xi. for 9852
        xii. vs 9295
        xiii. dlrs 8844
        xiv. it 8145
        xv. pct 7394
        xvi. 000 7140
        xvii. on 6597
        xviii. ; 6353
        xix. & 6304
        xx. lt 6302
3. Analyze the 20 most frequent types for each corpus. Based on this analysis, how is the newswire text (i.e., reuters.train.txt.gz) different from literature (i.e., gutenberg.txt. utf8.gz)?
    a. The “gutenberg.txt.utf8.gz” corpus contains more literary and general prose, resulting in common words like "he," "it," "was," and "not" being among the most frequent types.
    b. The double quotation mark (") appears frequently, indicating the inclusion of dialogue or quoted text in the corpus. This suggests the presence of dialogues or direct speech in the text.
    c. The presence of specific financial terms and symbols like "mln," "vs," "dlrs," "pct," "000," "&," and "lt" in “reuters.train.txt.gz” distinguishes it from the literary content in gutenberg.txt.utf8.gz.
    d. The presence of "&", “;” and "lt" suggests potential formatting and special character usage in newswire text i.e., < which stands for less than (<) sign in HTML.
    e. "gutenberg.txt.utf8.gz" represents a collection of literary works with diverse topics and genres, while "reuters.train.txt.gz" contains news and financial content with a specialized focus on current events and financial reporting.
4. Analyze the output of your tokenizer. Discuss the limitations of the simple approach to tokenization we implemented. Are there particular phenomena that are not tokenized correctly? Describe the kinds of errors that your tokenizer makes, and include supporting examples.
    a. Tokenization with the simple approach could not differentiate between a sequence of characters (words) and punctuation marks following the characters. For ex. ‘with:’, ‘for?,’ and ‘finish.’ were considered a single token.
    b. My tokenizer does not account for NLS/ multilingual characters and currencies other than $. Also, city names such as Saint John would be considered as 2 tokens and a person’s name Mr. Dusk would be considered as 3 separate tokens (Mr, . , Dusk). These are some of the inconsistencies I’ve noticed.
    c. < only makes sense in HTML if all these characters come together. My implementation does not support this. It splits that into 3 tokens as (&), (lt), and (;).
    d. “'but” is considered as a single token, but a single quote (‘) before but is lexically incorrect.