CMP711 Natural Language Processing

HW01 – Basic Text Processing & Language Models

Due Date: November 27, 2020 (Friday)

You have to write a Python program which will perform the following tasks:

- Your program should read a name of an input text file (.txt file) which contains an English text.
- Your program should find the sentences of the input text file and print the number of the sentences in the text. In order to find the end of the sentences, your program should assume that the sentences end with symbols *dot*, *question mark* or *exclamation mark.*
- Your program should tokenize the text in order to find the words of the text. A word is a string of letters. In order to normalize words, all words should be converted to lowercase. Your program should print the number of words (the number of total tokens) and the number of unique words (the size of the vocabulary).
- Your program should find unigram and bigram counts for the text and their unsmoothed probability values. Your program should print top 10 unigrams (unigrams with highest frequencies) together with their counts (and probabilities) and top 10 bigrams together with their counts (and probabilities).
- Then, your program should replace the three words with lowest frequencies with the special word UNK and it should assume that UNK represents all words that do not appear in the text. Your program should add UNK to the vocabulary of the text and remove three lowest frequency words from the vocabulary. Your program should re-compute bigram counts after the addition of UNK (and removal of the three lowest frequency words) and it should compute the smoothed bigram probabilities using Add-k smoothing method (assume that k=0.5). After finding smoothed probability values of all bigrams (including the bigrams containing UNK), you program should print again the smoothed probability values top 10 bigram values. Your program should also print the top 10 probability values of bigrams containing UNK.
- Your program should read a sentence from the keyboard and find the probability of that sentence using the smoothed probability values of bigrams. If the given sentence contains an unknown word, your program should assume that it is the special symbol UNK.

You should test your program with at least given three sample files (`hw01_tiny.txt`, `hw01_AMemorableFancy.txt`,`hw01_FireFairies.txt`) . You will submit the results produced for each file together with your program. You should create a result file for each test file (such as `hw01_tiny_Result.pdf`) to submit together with your program. The content of each result file should be as follows:

Number of Sentences in Test File: ….

Number of Total Tokens: ….

Number of Unique Words (Vocabulary Size): ….

Top 10 Unigrams with Highest Frequencies:
Unigram1 ItsCount ItsProbabibility
...

Top 10 Bigrams with Highest Frequencies:
Bigram1 ItsCount ItsProbabibility
...

After UNK addition and Smoothing Operations:
Top 10 Bigrams with Highest Frequencies:
Bigram1 ItsCount ItsProbabibility
...
Top 10 Bigrams with UNK:
Bigram1 ItsCount ItsProbabibility
...

SampleSentence1 ItsComputeProbability
SampleSentence2 ItsComputeProbability
...

Hand in:

You will submit your homework using the EVDEKAL environment. You have to upload the following four files.

- The source code (a Python program) of your homework. Put the source code of your program into a .txt file (`hw01.txt`) and upload this .txt file. **Make sure that this file contains only your Python program.**
- The result file (**`hw01_tiny_Result.pdf`**) containing the results of the file **`hw01_tiny.txt`** and the probability of at least two sentences.
- The result file (**`hw01_AMemorableFancy_Result.pdf`**) containing the results of the file **`hw01_AMemorableFancy.txt`** and the probability of at least two sentences.
- The result file (**`hw01_tiny_Result.pdf`**) containing the results of the file **`hw01_FireFairies.txt`** and the probability of at least two sentences.