Assignment II: Basic Text Processing

Introduction:

The goal of this assignment is to do some basic text processing task like stop words removal, stemming and lemmatizing the tokens generated by first assignment. We have to give output as word and its count per line.

Approach:

The output generated by first assignment i.e. tweet on one line followed by number of tokens in that tweet on second line and each individual token on each line is cleaned. The tweets and their token counts are removed from the file and the file containing token on each line is used as an input to the program.

I have followed modular approach while writing the program. I have used 'argparse' module which handles command-line arguments in a better way than conventional sys.argv. The program first checks validity of command line argument given i.e. name of the input file. After checking the validity of command-line arguments, the program tries to open input and output files. While opening it also checks for the 'IOError' and prints relevant error message. I have tried to handle the errors and exceptions in a best way possible.

If the files are opened successfully, it reads all the tokens and stores them in a list named 'tokens'. Tokens are passed to 'getLemmatizedTokens' function to get lemmas for each individual token. The parameter to 'getLemmatizedTokens' is a list of tokens in which each trailing newline as well as leading and trailing whitespaces are removed. 'getLemmatizedTokens' uses 'WordNetLemmatizer' to lemmatize tokens. Tokens which are canonical forms of either date or time are not lemmatized. They are kept as it is in a list of lemmatized tokens. All the digits, punctuation marks and stopwords are removed. Finally each word is lemmatized with the help of an object of 'WordNetLemmatizer' and corresponding part of speech tag. 'getLemmatizedTokens' returns a list of lemmatized tokens. The list of lemmatized tokens which is returned by 'getLemmatizedTokens' is passed as a parameter to 'getWordFrequency'. 'getWordFrequency' simply iterates over the list and creates an ordered dictionary having word as a key and word frequency as a value. Return value of 'getWordFrequency' is the ordered dictionary.

Finally each lemmatized token and its frequency is written to a file (each word, frequency pair on one line)

Observations:

This assignment depends heavily upon the accuracy of previous assignment. While processing a word lemmatizing works better than stemming. The reason behind the good accuracy of lemmatization is it maps word to its root form while stemming just cuts off affixes. A good example could be lemmatizing ‘ponies’ gives ‘pony’ while stemming the same word gives ‘poni’ which is not a meaningful word.

Discussions and References:

I discussed this assignment with Yashashree Kolhe. We discussed the strategy to solve the problem.

I referred official documentations of ‘argparse’ and ‘nltk’ module before writing the program.