TWEET TOKENIZATION & NORMALIZATION

Introduction:

The code takes the tokenized tweet (each token on a newline) as an input and generates the list of unique, lemmatized tokens along with their frequency (outputted as each token on a line with frequency of each token in front of the token) after ommiting some tokens (numbers, punctuation marks, stop words) which are not of much help while language processing.

Logic:

1) The tokens from the input dataset are retrieved.
2) If the token is a stop word, then omit that token. A list of stop words like ‘a’, ‘the’, ‘of’, etc. is taken from NLTK module.
3) If the token is any punctuation mark or number, omit such token.
4) If the token is a date or time, output the token as it is in the output file.
5) For other tokens, find the part-of-speech of the word and replace this token with its root form using lemmatization (Using lemmatize(word, pos) function in NLTK)

For eg: If ‘drunk’ is a token in the input, then the program identifies the pos of ‘drunk’ as V (i.e. Verb) and passes it as an arguement to lemmatize function -

lemmatize(‘drunk’, pos = ‘V’)

The above function will return ‘drink’as the root form and we will use ‘drink’ as the token instead of ‘drunk’ for further analysis.

This maps tokens like ‘drink’, ‘drunk’, ‘drinking’ to the same token ‘drink’ with frequency: 3.

6) Each unique token is stored in the dictionary. The dictionary stores token as key and the frequency of the token in the input dataset as value. If the token is not present in the dictionary, the entry {‘token’: 1} is added to the dictionary. Else, the value corresponding to the token is incremented by 1.

Observations:

It is observed that lemmatization is better than stemming in getting the root form of the word.
Eg: ‘drink’ is the root form of the word ‘drunk’, ‘drinking’.

Above case of ‘drunk’ -> ‘drink’ is not handled by stemming, since stemming only omits suffixes in order to get the root form.
Eg: Stemming will give the root form of verb ‘flies’ as ‘fli’ which is incorrect.

So I have used lemmatization so as to generate precise output.

Datasets used:

1) Fetched famous and most-liked tweets from wikipedia and prepared a dataset.
2) Added verbs, nouns in their different forms and verified whether the lemma is outputted precisely.

Resources:

1) Python Documentation of argparse
2) Python Documentation of NLTK
3) Online resources for Lemmatization