process_tweets_solution

March 18, 2023

1 Text Processing Exercise

In this exerise, you will learn some building blocks for text processing . You will learn how to normalize, tokenize, stemmeize, and lemmatize tweets from Twitter.

1.0.1 Fetch Data from the online resource

First, we will use the `get_tweets()` function from the `exercise_helper` module to get all the tweets from the following Twitter page https://twitter.com/AIForTrading1. This website corresponds to a Twitter account created especially for this course. This webiste contains 28 tweets, and our goal will be to get all these 28 tweets. The `get_tweets()` function uses the `requests` library and BeautifulSoup to get all the tweets from our website. In a later lesson we will learn how the use the `requests` library and BeautifulSoup to get data from websites. For now, we will just use this function to help us get the tweets we want.

```
In [1]: import exercise_helper

        all_tweets = exercise_helper.get_tweets()

        print(all_tweets)
```

```
['The Long-Term Stock Exchange Is Worth a Shot', 'Predicting Stock Performance with Natural Lang
```

1.0.2 Normalization

Text normalization is the process of transforming text into a single canonical form.

There are many normalization techniques, however, in this exercise we focus on two methods. First, we'll converting the text into lowercase and second, remove all the punctuation characters the text.

TODO: Part 1 Convert text to lowercase.

Use the Python built-in method `.lower()` for converting each tweet in `all_tweets` into the lower case.

```
In [2]: # your code goes here
```

```
        counter = 0

        for tweet in all_tweets:
            all_tweets[counter] = tweet.lower()
            counter += 1

        print(all_tweets)
```

```
['the long-term stock exchange is worth a shot', 'predicting stock performance with natural lang
```

Part 2 Here, we are using `Regular Expression` library to remove punctuation characters.

The easiest way to remove specific punctuation characters is with regex, the `re` module. You can sub out specific patterns with a space:

```
re.sub(pattern, ' ', text)
```

This will substitute a space with anywhere the pattern matches in the text.

Pattern for punctuation is the following `[^a-zA-Z0-9]`.

```
In [3]: import re

        counter = 0

        for tweet in all_tweets:
            all_tweets[counter] = re.sub(r'[^a-zA-Z0-9]', ' ', tweet)
            counter += 1

        print(all_tweets)
```

```
['the long term stock exchange is worth a shot', 'predicting stock performance with natural lang
```

1.0.3 NLTK: Natural Language ToolKit

NLTK is a leading platform for building Python programs to work with human language data. It has a suite of tools for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

Let's import NLTK.

```
In [4]: import os
        import nltk
        nltk.data.path.append(os.path.join(os.getcwd(), "nltk_data"))
```

TODO: Part 1 NLTK has `TweetTokenizer` method that splits tweets into tokens.

This make tokenizng tweets much easier and faster.

For `TweetTokenizer`, you can pass the following argument `(preserve_case= False)` to make your tokens in lower case. In the cell below tokenize each tweet in `all_tweets`

```
In [5]: from nltk.tokenize import TweetTokenizer

        #  your code goes here
        tknzr = TweetTokenizer(preserve_case= False)

        for tweet in all_tweets:
            print(tknzr.tokenize(tweet))
```

```
['the', 'long', 'term', 'stock', 'exchange', 'is', 'worth', 'a', 'shot']
['predicting', 'stock', 'performance', 'with', 'natural', 'language', 'deep', 'learning']
['comcast', 'acquiring', 'time', 'warner', 'cable', 'in', 'all', 'stock', 'deal', 'worth', '45',
['facebook', 'stock', 'drops', 'more', 'than', '20', 'after', 'revenue', 'forecast', 'misses']
['facebook', 'buying', 'whatsapp', 'for', '16b', 'in', 'cash', 'and', 'stock', 'plus', '3b', 'in
['netflix', 's', 'death', 'cross', 'is', 'the', 'third', 'for', 'faang', 'stocks', 'and', 'nasda
['after', 'yesterday', 's', 'signs', 'of', 'recovery', 'crypto', 'markets', 'see', 'drastic', 'l
['mf', 'sees', 'australia', 'risks', 'tilt', 'to', 'downside', 'on', 'china', 'trade', 'war']
['bitcoin', 'cash', 'clash', 'is', 'costing', 'billions', 'with', 'no', 'end', 'in', 'sight']
['sec', 'crypto', 'settlements', 'spur', 'expectations', 'of', 'wider', 'ico', 'crackdown']
['nissan', 's', 'drama', 'looks', 'a', 'lot', 'like', 'a', 'palace', 'coup']
['yahoo', 'finance', 'has', 'apparently', 'killed', 'its', 'api']
['tesla', 'tanks', 'after', 'goldman', 'downgrades', 'to', 'sell']
['goldman', 'sachs', 'to', 'open', 'a', 'bitcoin', 'trading', 'operation']
['tax', 'free', 'bitcoin', 'to', 'ether', 'trading', 'in', 'us', 'to', 'end', 'under', 'gop', 'p
['goldman', 'sachs', 'is', 'setting', 'up', 'a', 'cryptocurrency', 'trading', 'desk']
['robinhood', 'stock', 'trading', 'app', 'confirms', '110m', 'raise', 'at', '1', '3b', 'valuatio
['how', 'i', 'made', '500k', 'with', 'machine', 'learning', 'and', 'high', 'frequency', 'trading
['tesla', 's', 'finance', 'team', 'is', 'losing', 'another', 'top', 'executive']
['finance', 'sites', 'erroneously', 'show', 'amazon', 'apple', 'other', 'stocks', 'crashing']
['jeff', 'bezos', 'says', 'he', 'is', 'selling', '1', 'billion', 'a', 'year', 'in', 'amazon', 's
['us', 'government', 'commits', 'to', 'publish', 'publicly', 'financed', 'software', 'under', 'f
['the', 'dream', 'life', 'is', 'having', 'your', 'luggage', 'first', 'out', 'of', 'the', 'carous
['stocks', 'sink', 'as', 'apple', 'facebook', 'pace', 'the', 'tech', 'wreck', 'markets', 'wrap']
['elon', 'musk', 's', 'spacex', 'cuts', 'loan', 'deal', 'by', '500', 'million']
['nvidia', 'stock', 'falls', 'another', '12']
['anything', 'is', 'possible', 'in', 'this', 'world', 'exhibit', 'a', 'creation', 'of', 'a', 'se
['elon', 'musk', 'forced', 'to', 'step', 'down', 'as', 'chairman', 'tsla', 'short', 'all', 'the'
```

Part 2 NLTK adds more modularity for tokenization.

For example, stop words are words which do not contain important significance to be used in text analysis. They are repetitive words such as "the", "and", "if", etc. Ideally, we want to remove these words from our tokenized lists.

NLTK has a list of these words, `nltk.corpus.stopwords`, which you actually need to download through `nltk.download`.

Let's print out stopwords in English to see what these words are.

```
In [6]: from nltk.corpus import stopwords
        nltk.download("stopwords")
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/jdelgado/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[6]: True

1.0.4 TODO:

print stop words in English

In [7]:
```
# your code is here
print(stopwords.words('english'))
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll
```

TODO: Part 3 In the cell below use the `.split()` method to split each tweet into a list of words and remove the stop words from all the tweets.

In [8]:
```
## your code is here


for tweet in all_tweets:
    words = tweet.split()

    print([w for w in words if w not in stopwords.words("english")])
```

```
['long', 'term', 'stock', 'exchange', 'worth', 'shot']
['predicting', 'stock', 'performance', 'natural', 'language', 'deep', 'learning']
['comcast', 'acquiring', 'time', 'warner', 'cable', 'stock', 'deal', 'worth', '45', '2', 'billio
['facebook', 'stock', 'drops', '20', 'revenue', 'forecast', 'misses']
['facebook', 'buying', 'whatsapp', '16b', 'cash', 'stock', 'plus', '3b', 'rsus']
['netflix', 'death', 'cross', 'third', 'faang', 'stocks', 'nasdaq', 'composite', 'next']
['yesterday', 'signs', 'recovery', 'crypto', 'markets', 'see', 'drastic', 'losses']
['mf', 'sees', 'australia', 'risks', 'tilt', 'downside', 'china', 'trade', 'war']
['bitcoin', 'cash', 'clash', 'costing', 'billions', 'end', 'sight']
['sec', 'crypto', 'settlements', 'spur', 'expectations', 'wider', 'ico', 'crackdown']
['nissan', 'drama', 'looks', 'lot', 'like', 'palace', 'coup']
['yahoo', 'finance', 'apparently', 'killed', 'api']
['tesla', 'tanks', 'goldman', 'downgrades', 'sell']
['goldman', 'sachs', 'open', 'bitcoin', 'trading', 'operation']
['tax', 'free', 'bitcoin', 'ether', 'trading', 'us', 'end', 'gop', 'plan']
['goldman', 'sachs', 'setting', 'cryptocurrency', 'trading', 'desk']
['robinhood', 'stock', 'trading', 'app', 'confirms', '110m', 'raise', '1', '3b', 'valuation']
['made', '500k', 'machine', 'learning', 'high', 'frequency', 'trading']
['tesla', 'finance', 'team', 'losing', 'another', 'top', 'executive']
['finance', 'sites', 'erroneously', 'show', 'amazon', 'apple', 'stocks', 'crashing']
```

```
['jeff', 'bezos', 'says', 'selling', '1', 'billion', 'year', 'amazon', 'stock', 'finance', 'race
['us', 'government', 'commits', 'publish', 'publicly', 'financed', 'software', 'free', 'software
['dream', 'life', 'luggage', 'first', 'carousel', 'time']
['stocks', 'sink', 'apple', 'facebook', 'pace', 'tech', 'wreck', 'markets', 'wrap']
['elon', 'musk', 'spacex', 'cuts', 'loan', 'deal', '500', 'million']
['nvidia', 'stock', 'falls', 'another', '12']
['anything', 'possible', 'world', 'exhibit', 'creation', 'sequel', 'superbabies']
['elon', 'musk', 'forced', 'step', 'chairman', 'tsla', 'short', 'way']
```

1.0.5 Stemming

Stemming is the process of reducing words to their word stem, base or root form.

In the cell below, use the `PorterStemmer` method from the ntlk library to perform stemming on all the tweets

```
In [9]: from nltk.stem.porter import PorterStemmer

        # your code goes here

        for tweet in all_tweets:
            words = tweet.split()

            new_words = [w for w in words if w not in stopwords.words("english")]

            print([PorterStemmer().stem(w) for w in new_words])
```

```
['long', 'term', 'stock', 'exchang', 'worth', 'shot']
['predict', 'stock', 'perform', 'natur', 'languag', 'deep', 'learn']
['comcast', 'acquir', 'time', 'warner', 'cabl', 'stock', 'deal', 'worth', '45', '2', 'billion']
['facebook', 'stock', 'drop', '20', 'revenu', 'forecast', 'miss']
['facebook', 'buy', 'whatsapp', '16b', 'cash', 'stock', 'plu', '3b', 'rsu']
['netflix', 'death', 'cross', 'third', 'faang', 'stock', 'nasdaq', 'composit', 'next']
['yesterday', 'sign', 'recoveri', 'crypto', 'market', 'see', 'drastic', 'loss']
['mf', 'see', 'australia', 'risk', 'tilt', 'downsid', 'china', 'trade', 'war']
['bitcoin', 'cash', 'clash', 'cost', 'billion', 'end', 'sight']
['sec', 'crypto', 'settlement', 'spur', 'expect', 'wider', 'ico', 'crackdown']
['nissan', 'drama', 'look', 'lot', 'like', 'palac', 'coup']
['yahoo', 'financ', 'appar', 'kill', 'api']
['tesla', 'tank', 'goldman', 'downgrad', 'sell']
['goldman', 'sach', 'open', 'bitcoin', 'trade', 'oper']
['tax', 'free', 'bitcoin', 'ether', 'trade', 'us', 'end', 'gop', 'plan']
['goldman', 'sach', 'set', 'cryptocurr', 'trade', 'desk']
['robinhood', 'stock', 'trade', 'app', 'confirm', '110m', 'rais', '1', '3b', 'valuat']
['made', '500k', 'machin', 'learn', 'high', 'frequenc', 'trade']
['tesla', 'financ', 'team', 'lose', 'anoth', 'top', 'execut']
['financ', 'site', 'erron', 'show', 'amazon', 'appl', 'stock', 'crash']
['jeff', 'bezo', 'say', 'sell', '1', 'billion', 'year', 'amazon', 'stock', 'financ', 'race', 'sp
```

```
['us', 'govern', 'commit', 'publish', 'publicli', 'financ', 'softwar', 'free', 'softwar', 'licen
['dream', 'life', 'luggag', 'first', 'carousel', 'time']
['stock', 'sink', 'appl', 'facebook', 'pace', 'tech', 'wreck', 'market', 'wrap']
['elon', 'musk', 'spacex', 'cut', 'loan', 'deal', '500', 'million']
['nvidia', 'stock', 'fall', 'anoth', '12']
['anyth', 'possibl', 'world', 'exhibit', 'creation', 'sequel', 'superbabi']
['elon', 'musk', 'forc', 'step', 'chairman', 'tsla', 'short', 'way']
```

1.0.6 Lemmatizing

Part 1 Lemmatization is the process of grouping together the inflected forms of a word so they can be analyzed as a single item.

For reducing the words into their root form, you can use `WordNetLemmatizer()` method.

For more information about lemmatzing in NLTK, please take a look at NLTK documentation https://www.nltk.org/api/nltk.stem.html

If you like to understand more about Stemming and Lemmatizing, take a look at the following source: https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html

```
In [10]: nltk.download('wordnet') ### download this part
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     /Users/jdelgado/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

```
Out[10]: True
```

1.0.7 TODO:

In the cell below, use the `WordNetLemmatizer()` method to lemmatize all the tweets

```
In [11]: from nltk.stem.wordnet import WordNetLemmatizer

         # your code goes here

         for tweet in all_tweets:
             words = tweet.split()

             new_words = [w for w in words if w not in stopwords.words("english")]

             print([WordNetLemmatizer().lemmatize(w) for w in new_words])
```

```
['long', 'term', 'stock', 'exchange', 'worth', 'shot']
['predicting', 'stock', 'performance', 'natural', 'language', 'deep', 'learning']
['comcast', 'acquiring', 'time', 'warner', 'cable', 'stock', 'deal', 'worth', '45', '2', 'billio
['facebook', 'stock', 'drop', '20', 'revenue', 'forecast', 'miss']
['facebook', 'buying', 'whatsapp', '16b', 'cash', 'stock', 'plus', '3b', 'rsus']
```

```
['netflix', 'death', 'cross', 'third', 'faang', 'stock', 'nasdaq', 'composite', 'next']
['yesterday', 'sign', 'recovery', 'crypto', 'market', 'see', 'drastic', 'loss']
['mf', 'see', 'australia', 'risk', 'tilt', 'downside', 'china', 'trade', 'war']
['bitcoin', 'cash', 'clash', 'costing', 'billion', 'end', 'sight']
['sec', 'crypto', 'settlement', 'spur', 'expectation', 'wider', 'ico', 'crackdown']
['nissan', 'drama', 'look', 'lot', 'like', 'palace', 'coup']
['yahoo', 'finance', 'apparently', 'killed', 'api']
['tesla', 'tank', 'goldman', 'downgrade', 'sell']
['goldman', 'sachs', 'open', 'bitcoin', 'trading', 'operation']
['tax', 'free', 'bitcoin', 'ether', 'trading', 'u', 'end', 'gop', 'plan']
['goldman', 'sachs', 'setting', 'cryptocurrency', 'trading', 'desk']
['robinhood', 'stock', 'trading', 'app', 'confirms', '110m', 'raise', '1', '3b', 'valuation']
['made', '500k', 'machine', 'learning', 'high', 'frequency', 'trading']
['tesla', 'finance', 'team', 'losing', 'another', 'top', 'executive']
['finance', 'site', 'erroneously', 'show', 'amazon', 'apple', 'stock', 'crashing']
['jeff', 'bezos', 'say', 'selling', '1', 'billion', 'year', 'amazon', 'stock', 'finance', 'race'
['u', 'government', 'commits', 'publish', 'publicly', 'financed', 'software', 'free', 'software'
['dream', 'life', 'luggage', 'first', 'carousel', 'time']
['stock', 'sink', 'apple', 'facebook', 'pace', 'tech', 'wreck', 'market', 'wrap']
['elon', 'musk', 'spacex', 'cut', 'loan', 'deal', '500', 'million']
['nvidia', 'stock', 'fall', 'another', '12']
['anything', 'possible', 'world', 'exhibit', 'creation', 'sequel', 'superbabies']
['elon', 'musk', 'forced', 'step', 'chairman', 'tsla', 'short', 'way']
```

TODO: Part 2 In the cell below, lemmatize verbs by specifying `pos`. For `WordNetLemmatizer().lemmatize` add `pos` as an argument.

```
In [12]: from nltk.stem.wordnet import WordNetLemmatizer

         # your code goes here

         for tweet in all_tweets:
             words = tweet.split()

             new_words = [w for w in words if w not in stopwords.words("english")]

             print([WordNetLemmatizer().lemmatize(w, pos='v') for w in new_words])
```

```
['long', 'term', 'stock', 'exchange', 'worth', 'shoot']
['predict', 'stock', 'performance', 'natural', 'language', 'deep', 'learn']
['comcast', 'acquire', 'time', 'warner', 'cable', 'stock', 'deal', 'worth', '45', '2', 'billion'
['facebook', 'stock', 'drop', '20', 'revenue', 'forecast', 'miss']
['facebook', 'buy', 'whatsapp', '16b', 'cash', 'stock', 'plus', '3b', 'rsus']
['netflix', 'death', 'cross', 'third', 'faang', 'stock', 'nasdaq', 'composite', 'next']
['yesterday', 'sign', 'recovery', 'crypto', 'market', 'see', 'drastic', 'losses']
['mf', 'see', 'australia', 'risk', 'tilt', 'downside', 'china', 'trade', 'war']
['bitcoin', 'cash', 'clash', 'cost', 'billions', 'end', 'sight']
```

```
['sec', 'crypto', 'settlements', 'spur', 'expectations', 'wider', 'ico', 'crackdown']
['nissan', 'drama', 'look', 'lot', 'like', 'palace', 'coup']
['yahoo', 'finance', 'apparently', 'kill', 'api']
['tesla', 'tank', 'goldman', 'downgrade', 'sell']
['goldman', 'sachs', 'open', 'bitcoin', 'trade', 'operation']
['tax', 'free', 'bitcoin', 'ether', 'trade', 'us', 'end', 'gop', 'plan']
['goldman', 'sachs', 'set', 'cryptocurrency', 'trade', 'desk']
['robinhood', 'stock', 'trade', 'app', 'confirm', '110m', 'raise', '1', '3b', 'valuation']
['make', '500k', 'machine', 'learn', 'high', 'frequency', 'trade']
['tesla', 'finance', 'team', 'lose', 'another', 'top', 'executive']
['finance', 'sit', 'erroneously', 'show', 'amazon', 'apple', 'stock', 'crash']
['jeff', 'bezos', 'say', 'sell', '1', 'billion', 'year', 'amazon', 'stock', 'finance', 'race', '
['us', 'government', 'commit', 'publish', 'publicly', 'finance', 'software', 'free', 'software',
['dream', 'life', 'luggage', 'first', 'carousel', 'time']
['stock', 'sink', 'apple', 'facebook', 'pace', 'tech', 'wreck', 'market', 'wrap']
['elon', 'musk', 'spacex', 'cut', 'loan', 'deal', '500', 'million']
['nvidia', 'stock', 'fall', 'another', '12']
['anything', 'possible', 'world', 'exhibit', 'creation', 'sequel', 'superbabies']
['elon', 'musk', 'force', 'step', 'chairman', 'tsla', 'short', 'way']
```

In [ ]: