Tweetanalyzer.py

Overview

The delivered .zip-file contains the following structure:

src

- *__init__.py*
- *calculator.py*
  - Calculating unit of the tweetanalyzer
- *data_loader.py*
  - Loads and cleans the input-data
- *tweetanalyzer.py*
  - Main module for running the script
- *tweets.txt*
  - Raw Data

test

- *hashtags.txt*
  - Dummy data for testing purposes
- *lower_letters.txt*
  - Dummy data for testing purposes
- *special_characters.txt*
  - Dummy data for testing purposes
- *test_data_loader.py*
  - Unit tests for data_loader.py

output

- *unigrams.txt*
  - 500 common unigrams, with stopwords
- *bigrams.txt*
  - 100 common bigrams, with stopwords
- *trigrams.txt*
  - 100 common trigrams, with stopwords

output_without_stopwords

- *unigrams.txt*
  - 500 common unigrams, without stopwords
- *bigrams.txt*
  - 100 common bigrams, without stopwords
- *trigrams.txt*
  - 100 common trigrams, without stopwords
- *cleaneddata.txt*
  - Temporary (cleaned) file without stopwords

Processing Algorithm

1. Load the .txt-file
2. Data Cleaning
    a. Decode to UTF-8
    b. Remove html tags
    c. Lower all letters for standardization
    d. Remove apostrophes (e.g. replace “I’m” by “I am”)
    e. (optional) remove stopwords
    f. Remove links
    g. Remove hashtags
    h. Remove special characters
    i. Print in a temporary file
3. Calculating the n-grams
    a. Load the temporary file
    b. Calculate Unigrams and print in new target file
    c. Calculate Bigrams and print in new target file
    d. Calculate Trigrams and print in new target file
    e. Remove the temporary file
4. Run tests

Before you start

- Please make sure you’re using Python 2.7 (the script may not work on other python versions)
- If you have not already installed the NLTK-module (Natural language toolkit) for python, please follow these steps (Linux Mint (Debian) installation):
    - Open the shell and enter: Sudo pip install –U nltk
    - Start python in the shell and enter the following commands:
        - *Import nltk*
        - *nltk.download(‘punkt’)*

If you’re using Windows you can find an installation guide here: http://www.nltk.org/install.html

How to start

- Start the program by typing “*python tweetanalyzer.py*” in the shell. (The tweetanalyzer.py-file is the “main”-file)
- If you want to start the tests, you’ve to start them manually by entering “*python test_data_loader.py”* in the shell.

Cool modifications

After you’ve started the script, it will create three .txt files in the src-folder. You can modify the tweetanalyzer.py-file by changing the number of printed n-grams. (You may change it to tetra- or pentagrams☺)

In addition, I’ve implemented a functionality which sorts out the so-called “stopwords”. Unfortunately, the script took about 15-20 Minutes to finish. That’s why I’ve commented the line out. You can enable the functionality by uncommenting the marked line in the data_loader.py. (see below)

```
html_parser = HTMLParser.HTMLParser()
APPOSTROPHES = {"'s":"is", "'re":"are", "'m":"am", "'t":"not", "'ll":"will", "ill":"I will", "im":"I am",
                "wanna":"want to", "gonna": "going to"}

def cleanData(rawTweets, outputfile):
 """
 Cleans the given Data according to certain text mining methods and prints the cleaned output in a textfile
 :param rawTweets: Input source which has to be cleaned
 :param outputfile: Name the ouput file should have
 :return:
 """
 NewFile = open(outputfile, 'w')
 for eachline in rawTweets:
  # decode to UTF-8
  cleanedTweets = eachline.decode("utf8").encode('ascii', 'ignore')
  # remove html tags
  cleanedTweets = html_parser.unescape(cleanedTweets)
  # lower all letters for standardization
  cleanedTweets = cleanedTweets.lower()
  # remove apostrophes and standardize data
  cleanedTweets = [APPOSTROPHES[word] if word in APPOSTROPHES else word for word in cleanedTweets.split()]
  cleanedTweets = " ".join(cleanedTweets)
  # remove stopwords
  cleanedTweets = ' '.join([word for word in cleanedTweets.split() if word not in stopwords.words("english")])
  # remove links
  cleanedTweets = re.sub(r'http\S+', '', cleanedTweets) #remove links
  # remove hasthags
  cleanedTweets = re.sub(r'#\w+ ?', '', cleanedTweets)
  # remove remaining special characters
  cleanedTweets = re.sub('[^A-Za-z0-9 ]+', '', cleanedTweets)
  # print the output in a new File
  NewFile.write('%s \n' % cleanedTweets)
```