Solution 1:

Wendys\_tweet\_scan.py

* Used “twitter” API by generating Consumer and Token keys on twitter account to create an App to gather data. These keys are used to authenticate first so as to gain access to twitter data. “twitter\_api” variable stores a 12-digit hex value that indicates whether or not access is authenticated or not.
* “twitter.com/wendys” is the default link where all the tweets with “@wendys” handle can be read.
* The keywords are being matched using Regular Expressions (re) in python.
* Count of each good reviews (g\_tags) and bad reviews (b\_tags) is recorded each time a tweet is scanned. All the tweets are enclosed into “<p>” tags.
* Ultimately, results are stored in “wendy\_rev.xls”

Solution 2:

Wendys\_tweet\_collection.py

* Used “twitter” API by generating Consumer and Token keys on twitter account to create an App to gather data. These keys are used to authenticate first so as to gain access to twitter data. “twitter\_api” variable stores a 12-digit hex value that indicates whether or not access is authenticated or not.
* “json\_statuses\_file.txt” file is created. Here, all the JSON that involve tweets/statuses information will be stored later on using “json.dumps”
* “query” is used to scrape twitter for a keyword. Here we are looking for Wendys tweets, hence “@wendys” is used. The most recent tweets are taken. Max tweets count is taken as 100 as twitter won’t allow to scrape all the tweets present.
* The “wendys\_tweets” will gather a lot of unnecessary data out of which we only need to keep “wendys\_tweets[‘statuses’]” data relevant to tweets text only.
* These ‘statuses’ are then stored in text file mentioned above.

Wendeys\_pos\_neg.py

* Here, the JSON file is opened in read mode to read the JSON.
* “status\_texts” will be used to reach “texts” inside the JSON tree. “statuses” is used to store all the individual words in these texts/tweets.
* Here, only a limited positive and negative words’ list is used. More sophisticated list would yield better results. Such dictionaries are available online. Or if the company already has such keywords/phrases, we can use them to create optimized results.
* Using these two lists to match with the statuses in for loop, two for loops are nested inside it to check for good and bad reviews. At presence of each positive or negative review match, corresponding count is increased. The same is saved in “wendy\_rev.xls”