# This is a Python program to find out negative and positive tweets about Wendys

import twitter

import json

from urllib import unquote

#---------------------------------------------

# Function to authenticate using Twitter API. The keys used are critical to one user and are not #meant to be shared. Later, while scraping twitter for CL Kings, I might need company specific #twitter account to create such an app.

def oauth\_login():

CONSUMER\_KEY = 'SKeR7JceYHh6V7MA9aQg2LIs9'

CONSUMER\_SECRET ='YkMQ6GY1EuOVwxfHnc1lc8wr6OzoIveYiKgdbBwqtPyillJj41'

OAUTH\_TOKEN = '289922127-yer28rcHdG14s6REh9z2LZ53ibBgLnsy6djB0Ff8'

OAUTH\_TOKEN\_SECRET = 'B7eGgOfnxE7nGRK25BPanc6rj9AoToR6ENb3bdsKYkFx9'

auth = twitter.oauth.OAuth(OAUTH\_TOKEN, OAUTH\_TOKEN\_SECRET,

CONSUMER\_KEY, CONSUMER\_SECRET)

twitter\_api = twitter.Twitter(auth=auth)

return twitter\_api

#----------------------------------------------

# Main body of the program

print 'This is a Python program to find out positive and negative tweets about Wendys'

twitter\_api = oauth\_login()

# File to store JSON data. File will be later used to parse JSON to find positive and negative #reviews

json\_statuses\_file = open('json\_statuses\_file.txt', 'w')

query = "@wendys" # The query string

recent = "recent" # The type of result we want, this will give only recent tweets

tweet\_count = "100" # Search API limits maximum number of tweets returned to 100

# Use the Twitter Search API to get the results. The query is used to get 100 recent tweets with the word "@Wendys".

wendys\_tweets = twitter\_api.search.tweets(q=query, count=tweet\_count, include\_entities=1, result\_type=recent)

twitter\_statuses = wendys\_tweets['statuses']

# Iterate to generate 99 queries. In each query we try to get 100 tweets. But API might return less #than 100 tweets sometimes. Here, I have just created sample dictionary of positive and negative #keywords for sentiment analysis, it will return very few results/tweets.

for page\_counter in range(0, 99):

try: # Find the smallest Twitter ID in the set of statuses returned for each query

no\_of\_statuses = len(wendys\_tweets['statuses'])

for status\_counter in range(0, no\_of\_statuses):

tweet\_ids.append(wendys\_tweets['statuses'][status\_counter]['id'])

min\_tweet\_id = min(tweet\_ids)

except: # Exception handling

print "Exception encountered. Exiting for-loop."

break; # Exit the iteration

# Returns results with an ID less than (that is, older than) or equal to the specified ID.

wendys\_tweets = twitter\_api.search.tweets(q=query, count=tweet\_count, include\_entities=1, result\_type=recent, max\_id=min\_tweet\_id)

twitter\_statuses = twitter\_statuses + wendys\_tweets['statuses']

no\_of\_statuses = len(wendys\_tweets['statuses'])

print >> execution\_output\_file, "No. of statuses now - ", len(twitter\_statuses)

print 'Total no.of statuses gathered - ', len(twitter\_statuses)

print 'Printing statuses to file....'

json\_statuses\_file.write(json.dumps(twitter\_statuses, indent=1))

json\_statuses\_file.close()

execution\_output\_file.close()

print 'Printing to file complete.'

print 'The program has completed.'