Assignment 3

DOCUMENT RETRIEVAL SYSTEM

Brief

- **Due date : 10/23/2016 11:59 pm**
- **Data**: fetch_tweets.py, Writeup
- **Handin**: follow the handin instruction.

In this homework you will implement a scaled down version of a document retrieval system. Precisely, You will try to implement the following functionalities - **retrieve relevant documents to a search query** and **try to cluster similar documents.** For this homework, we will use Twitter Data as our Dataset. Each Tweet is considered as a document and you are supposed to answer query 1 and query 2 on the tweets(considered as documents). You will use **python3** together with Scikit Learn package to perform the tasks.

Setup

- sklearn. If you machine does not contain such package, run the following command to get the latest package:
  - `sudo pip3 install -U scikit-learn`
- oauth2. If you machine does not contain such package, run the following command to get the latest package:
  - sudo pip3 install -U oauth2
- Note that you are not allowed to use any outside libraries for the whole hw3 except the ones which have been provided in the given python scripts.

To access the Twitter API, you will need to setup a Twitter Developer account.

- Create a twitter account if you do not already have one.
- Go to https://dev.twitter.com/apps and log in with your twitter credentials.

Click "Create new application"

- Fill out the form and agree to the terms. Put in a dummy website (http://localhost.com, for example) if you don't have one you want to use.

- On tab "Keys and Access Tokens", Click "Create my access token". You can Read more about Oauth authorization.
- Open **fetch_tweets.py** and set the variables corresponding to the consumer key, consumer secret, access token, and access secret.

```
access_token_key = "<Enter your access token key here>"
access_token_secret = "<Enter your access token secret here>"
consumer_key = "<Enter consumer key>"
 consumer_secret = "<Enter consumer secret>"
```

Twitter API

Search API

To get the tweets related to a search term, you are supposed to send a GET request to https://api.twitter.com/1.1/search/tweets.json. (This is what happens generally, but we have done this for you.. Run the following command to mimic the process). To learn more about this, visit https://dev.twitter.com/docs/api/1.1/get/search/tweets

Run the following to get the tweets related to a term of your choice

```
$ python3 fetch_tweets.py -c fetch_by_terms -term "[your_chosen_term]" >
search_output.txt
```

Turn-In

Choose a search term which is appeared in at least 100 tweets. Handin **search_output_[your NetID].txt**. In your report (reporta_[your NetID].txt), include your search term and some example tweets. Some examples of search term are presidential elections, Game of Thrones, etc.

When working with the text data, most of the applications demand the usage of building feature vectors from the text documents. So, in this assignment, we will be using the following three methods for building feature vectors.

1. **TFIDF Vectorizer:** http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html#sklearn.feature_extraction.text.TfidfVectorizer
2. **Count Vectorizer:** http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html#sklearn.feature_extraction.text.CountVectorizer
3. **Hashing Vectorizer:** http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.HashingVectorizer.html#sklearn.feature_extraction.text.HashingVectorizer

Query:

You will use the tweets obtained from Twitter data to answer the following queries:

1. Given a query(some search topic of your choice), return top 10 similar tweets to the given query.

2. Return 5 clusters of similar tweets(Here, a cluster is a set of tweets that are similar).

FAQ:

1. Should the Data be cleaned?

A: Yes. Data Cleaning is a necessary step in the pipeline(Tokenization, stop word removal, etc).

2. There are lots of hyper parameters in each of the Vectorizers, which ones are relevant to the task at hand?

A: min_df, max_df are some parameters of interest. You are welcome to experiment other parameters.

Bonus Points:

Generally, unigrams are used as features. Experiment with other types of features like bigrams, trigrams, Part of Speech tags, combinations of these. Document whatever you have tried. If you are interested, we can discuss about this during the office hours of instructors(Professor/ TA).

What to Turn-In:

1. Your python file - [your-netid].py
2. A Comparison of the performance of the aforementioned vectorizers. Justify your answer!
3. README file

Evaluation Criteria:

All the students will be evaluated on a common dataset(set of tweets). So ensure that your submission is as generic as possible! No Hard Coding!!!!

Cheating/Plagiarism Policy:

Any cheating may result in a **fail grade** in the course **and/or be reported to the university higher authorities for appropriate action**.