E6998 – HW#3

Due 11/6/2017 at 11:59PM (23:59) - EST

Twitter API

Classifier

Crawler

mongo

.txt

Using the code from the document classifier (HW#1), streaming twitter engine (HW#2) and the last class ([here](https://github.com/phoulihan/columbia/blob/master/class/classEx/crawler.py)), build a streamlined system that contains the following functionality:

1. ***Crawler (50-points):***

You will have a crawler that accepts in an array with the following strings, each representing a topic:

topics = [“*politics”*, “*astronomy”*, “*medical”*, “*music”, “sports”]*

The crawler is to crawl at least 50 documents per topic, write the text files to a directory names after each topic. You need to have logic that rejects documents that are less than 50 words, meaning, you will only write crawled content to a text file if it contains at least 50 words. Your code needs to auto-create the folders (hint: import os).

1. ***Classifier (10 points):***

Using the documents from the crawl, train a model that can be called up to classify arbitrary text into any of the 5 topics.

1. ***Streaming Twitter (10 points):***

Use the following strings in your streaming twitter app:

setTerms = [“potus”, “moon and the sun”, “pharmacy”, “drake”, “quarterback”]

You will call up your model and classify (*topic* variable) the bodies of tweets and store the following in a mongo database:

*followers*, *screen\_name*, *friends\_count*, *created\_at*, *message\_id*, *location, topic*

Export a sample of a test run to a json file called *hw3.json*.

1. ***Streamline the entire process (30-points)***

You need to streamline your code, where one execution of code will run through the entire process: crawl 🡪 train 🡪 Twitter API 🡪 Classify bodies of tweet 🡪 write to mongo

The code is to keep the API running, continuously classify tweets and write to mongo

**Push all python code and the exported json file to your github accounts. Anyone with their own path name and twitter app tokens should be able to run steps a-d. Any steps that fail to run will receive 0 points. There will be no partial credit for code that fails to work.**