OLD DOMINION UNIVERSITY

CS 495: Introduction to Web Science
Instructor: Micheal L. Nelson, Ph.D
Fall 2014 Thursdays 4:20pm – 7:10pm ECSB 2120

Assignment # 10
Joseph Elder UIN: 00844802

Honor Pledge
I pledge to support the Honor System of Old Dominion University. I will refrain from any form of academic dishonesty or deception, such as cheating or plagiarism. I am aware that as a member of the academic community it is my responsibility to turn in all suspected violations of the Honor Code. I will report to a hearing if summoned

December 10, 2014

1.1 Question 1

1.1.1 The Problem

1. Choose a blog or a newsfeed (or something similar as long as it has an Atom or RSS feed). It should be on a topic or topics of which you are qualified to provide classification training data. In other words, choose something that you enjoy and are knowledgable of. Find a feed with at least 100 entries.

Create between four and eight different categories for the entries in the feed:

examples:

work, class, family, news, deals

liberal, conservative, moderate, libertarian

sports, local, financial, national, international, entertainment

metal, electronic, ambient, folk, hip-hop, pop

Download and process the pages of the feed as per the week 12 class slides.

1.1.2 The Solution

The python version 2.7.6 code for Assignment 10 is located in the file 'A10.py'. Being that I am new to using RSS and Atom feeds I did not have any good feeds to use at first for this assignment. I first had to locate a feed which would be of interest to me, I used an RSS feed search engine to find several feeds which I found potentially interesting. The feed I chose is titled 'Uploads by Top Gear' and it can be found at the following web address.

http://gdata.youtube.com/feeds/base/users/TopGear/uploads?alt=rss&v=2&orderby=published&client=ytapi-youtube-profile

I chose this RSS feed because I enjoy watching Top Gear and this particular feed contained only video clips from the UK television series. Additionally there were well over 100 entries in the feed so it served as a decent choice for all intended purposes. After looking up several of the feed entries and using my general Top Gear knowledge I was able to come up with the following categories to be used to classify each feed entry: “Car Review”,“Adventure”, “Celebrity Driver”, “Special Challenge”, “Races/Racing/Laps/Laptime”, “The News/Meta”. The categories for the most part are easy to distinguish per entry and are loosely based off of the parts of the show itself.

The first category “Car Review” is for the entries where the hosts of the show are reviewing a specific car and giving specifications along with their personal prejudices. Sometimes on the episodes all three hosts will review similar cars and have to put them through several challenges so determination between the categories “Car Review” and “Special Challenges” was difficult for several entries. The rule of thumb for selection came down to if the video contained specifications about the car model it would be a review, if the video contained only a challenge the car was being put through it would be considered a challenge. The category “Adventure” classifies all of the videos where the hosts are taking cars on a road trip to some destination. The “Celebrity Driver” category consists almost entirely of entries about the 'Star in a Reasonably Priced Car' bit on the tv show where each week a celebrity will race around a test track competing for the best time to be put on a board with everyone else's times. The “Race/Racing/Laps/Laptime” categories is for entries which pertain to professional racing, racers, the professional test driver “the stig” setting test laptimes, Nuremburg laps, etc. The final categories is for entries about car news which is a regular segment on the show, videos about the hosts or behind the scenes will mostly fall into the “Meta” category shared with the news.

The feed is parsed using a python module called 'feedparser' which aids in accessing all of the information the RSS feed has to offer easily in python. The first 150 entries are saved, 150 are used to accommodate for several broken links where no content was available. The manual rating process will collect the first 50 ratable entries from the feed.

1.2 Question 2

1.2.1 The Problem

2. Manually classify the first 50 entries, and then classify (using
the fisher classifier) the remaining 50 entries. Report the cprob()
values for the 50 titles as well. From the title or entry itself,
specify the 1-, 2-, or 3-gram that you used for the string to
classify. Do not repeat strings; you will have 50 unique strings.
For example, in these titles the string used is marked with *s:

Rachel Goswell - "Waves Are Universal" (LP Review)
The *Naked and Famous* - "Passive Me, Aggressive You" (LP Review)
Negativland - "Live at Lewis's, Norfolk VA, November 21, 1992" (concert)
Negativland - "*U2*" (LP Review)

Note how "Negativland" is not repeated as a classification string.

Create a table with the title, the string used for classification,
cprob(), predicted category, and actual category.

1.2.1 The Solution

Python is used to read each entry in 'entries.txt', a loop is iterated through until all 100 entries have been classified under one of the 6 categories represented by the characters A through F. After collecting the manual ratings into the file 'rated.txt', the first 50 titles from the entries are used to train the classifier with the manual ratings. Unfortunately the rest of the solution is still in development at the moment.

References

1. https://docs.python.org/2/

2. http://www.pythonforbeginners.com/feedparser/using-feedparser-in-python

3. http://gdata.youtube.com/feeds/base/users/TopGear/uploads?alt=rss&v=2&orderby=published&client=ytapi-youtube-profile

4. http://ctrlq.org/rss/