Bryan Carey

CS432 Assignment #9

Due: 11:59pm May 1 2017

1. Choose a blog or a newsfeed (or something similar with an Atomor RSS feed). Every student should do a unique feed, so please "claim" the feed on the class email list (first come, first served). It should be on a topic or topics of which you are qualified to provide classification training data. Find something with at least 100 entries (or items if RSS).

http://top10000s.blogspot.com/feeds/posts/default?max-results=110

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

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

Be sure to upload the raw data (Atom or RSS) to your GitHub account.

Create a table with 100 rows. This is your "ground truth" (or "gold standard") data.

To extract the blog titles I used grabTitle.py, which was created after a conversation and with some guidance through the assignment by a classmate. Accessing the feed was similar to handling other data as we have through semester so looking through my old .py files provided a nice reference.

```
import feedparser
import re
rss_url = "http://top10000s.blogspot.com/feeds/posts/default?max-results=110"

feed = feedparser.parse(rss_url)

for post in feed.entries:
    titles = post.title
    pTitles = re.sub('[0-9].','',titles)
    pTitles2 = re.sub('\.', '', pTitles)
    print pTitles2
```

The content generated is as follows-(The full 100 titles are stored in dataTitles.txt)

```
Outkast: Hey Ya!
Gorillaz: Feel Good, Inc
Missy Elliot: Get Ur Freak On
Gnarls Barkley: Crazy
*NSYNC: Bye Bye Bye
Nelly: Hot In Herrre
Beyonce (feat Jay-Z): Crazy In Love
Kelly Clarkson: Since U Been Gone
Dixie Chicks: Not Ready To Make Nice
Alicia Keys: Fallin'
Modest Mouse: Float On
Jason Mraz: The Remedy (I Won't Worry)
The Postal Service: Such Great Heights
U Beautiful Day
Yeah Yeah Yeahs: Maps
```

Bryan Carey

CS432 Assignment #9

Due: 11:59pm May 1 2017

I classified each of the songs collected from the blog according to one of 5 categories: Rock, Pop, Hip-Hop, R&B, or Other. The classifications I assigned are stored in dataClassification.txt.

Following the creation of grabTitle.py I constructed getContent.py. This process was completed separately as the content of each (Titles & Content) are each parsed differently. The data would later be recombined. I studied the code snippets provided at http://stackoverflow.com/questions/13748674/python-re-sub-how-to-substitute-all-u-or-us-with-you to re-familiarize myself with the use of 'sub' to get rid of unwanted characters.

```
import feedparser
import re

rss_url = "http://top10000s.blogspot.com/feeds/posts/default?max-results=110"

feed = feedparser.parse(rss_url)

contents = []
count = 0
for entry in feed.entries:
    if 'content' in entry:
        content = entry.content

    contents.append(entry.content)

subList = contents[:100]

for content in subList:
    count += 1
    print '\n'

    parse1 = re.sub('<[^>]*>', ' ', str(content))
    parse2 = parse1[126:]
    parse3 = re.sub("\\\\",'', parse2)
    parse4 = re.sub("  ",' ', parse3)
    parse5 = re.sub("  ", ' ', parse4)
    parse6 = re.sub("nbsp",'', parse4)

    print '\n'
    print parse5
```

A snippet from the generated content follows

Once I had both the titles and content from each of the entries, I merged them together with combineData.py, which was created through code from the PCI text, and the examples provided in class lecture slides.

Bryan Carey

CS432 Assignment #9

Due: 11:59pm May 1 2017

```
def parsing(summary,title):
    global numEntries
    print '\n'
    numEntries = numEntries + 1

    print numEntries
    print title
    print summary
    print '\n'

    substitute_1= re.sub('<[^>]*>',' ', summary)
    substitute_2= re.sub('   Continue reading...','',substitute_1)
    substitute_3= re.sub('  ',' ',substitute_2)

    #modified from grabContent.py
    omit_1 = title+" "+substitute_3
    omit_2 = re.sub("'","\\\\'",omit_1)
    omit_3 = re.sub('"','',omit_2)
    omit_4 = re.sub("`",'',omit_3)
    omit_5 = re.sub("[0-9]+\.", '', omit_4)
    print omit_5
    return title,substitute_3,omit_5
    print '\n'

#taken from class  PowerPoint
for e in data.entries:

    if 'summary' in e:
        summary = e.summary
    else:
        summary = e.description


    parsedtitle, parsedsummary, combinedsumtitle = parsing(summary,e.title)

    outFile1.write(parsedtitle)
    outFile1.write('\n')

    outFile2.write(parsedsummary)
    outFile2.write('\n')

    outFile3.write(combinedsumtitle)
    outFile3.write('\n')
```

The output of the file is as follows-

```
Outkast: Hey Ya!   Found on: Speakerboxx / The Love Below (2003) Picked by: RN, MT, TF, SL, KT, WA, PA Why would
Gorillaz: Feel Good, Inc.   Found on: Demon Days (2005) Picked by: RN, BG, MT, SL, PA, WA Gorillaz weren\'t a re
Missy Elliot: Get Ur Freak On   Found on: Miss E...So Addictive (2001) Picked by: RN, MT, TF, SL, KT, WA For fou
Gnarls Barkley: Crazy   Found on: St. Elsewhere (2006) Picked by: MT, TF, SL, KT, WA, PA Lovely neo gospel pop f
*NSYNC: Bye Bye Bye   Found on: No Strings Attached (2000) Picked by: KT, RN, SL, SVH, WA, PA As loathe as I am
```

Bryan Carey

CS432 Assignment #9

Due: 11:59pm May 1 2017

& So forth. I then checked the content against the original blog contents, and the data checked out. I then created a table in which to store the data. Here is a snippet from my Q1 table.

	Title	Genre
2	Outkast: Hey Ya!	Hip-Hop
3	Gorillaz: Feel Good, Inc	Rock
4	Missy Elliot: Get Ur Freak On	Hip-Hop
5	Gnarls Barkley: Crazy	Hip-Hop
6	*NSYNC: Bye Bye Bye	Pop
7	Nelly: Hot In Herrre	Hip-Hop
8	Beyonce (feat Jay-Z): Crazy In Love	Hip-Hop
9	Kelly Clarkson: Since U Been Gone	Pop
10	Dixie Chicks: Not Ready To Make Nice	Other
11	Alicia Keys: Fallin'	R&B
12	Modest Mouse: Float On	Rock
13	Jason Mraz: The Remedy (I Won't Worry)	Rock
14	The Postal Service: Such Great Heights	Rock
15	U Beautiful Day	Rock

2. Train the Fisher classifier on the first 50 entries (the "training set"), then use the classifier to guess the classification of the next 50 entries (the "test set").

Create a table with 50 rows

Assess the performance of your classifier in each of your categories by computing precision, recall, and F-measure. Use the "macro-averaged" label based method, as per:

http://stats.stackexchange.com/questions/21551/how-to-compute-precision-recall-for-multiclass-multilabel-classification

For this portion of the assignment we pulled from docclass.py, found at https://github.com/arthur-e/Programming-Collective-Intelligence/blob/master/chapter6/docclass.py and the examples provided by Dr. Nelson in his lecture slides once again. I then created trainer.py. Here is the body of the code

Bryan Carey

CS432 Assignment #9

Due: 11:59pm May 1 2017

```
trainingCount90 = 90
maxTrainingData = 100
remainingClassifications = 0


for entry in input:
  categories.append(entry)
for entry in input_2:
  summaryTitles.append(entry)

classifier = docclass.fisherclassifier(docclass.getwords)

#taken from lecture PowerPoint
classifier.setdb('mln.db')

for entry in summaryTitles:
    if count<trainingCount50:
     docclass.train(classifier, entry, categories[count])
     count += 1


w=maxTrainingData - count
intialpredictions = maxTrainingData - count

while count<maxTrainingData:
  print count
  prediction=classifier.classify(summaryTitles[(count)])
  predictions.append(prediction)
  count+=1

if len(predictions) > 49:
   output1=open('Classify50.txt','w')
if len(predictions) < 11:
   output2= open('Classify10.txt','w')

print(len(predictions))
for prediction in predictions:
   if len(predictions)>49:

       output1.write(prediction)
   if len(predictions)<11:

       output2.write(prediction)

```

Bryan Carey

CS432 Assignment #9

Due: 11:59pm May 1 2017

Trainer.py was used to classify the last 50 pieces of data based upon the classifications I passed into the trainer. The predictions of the trainer are stored in Classify50.txt. I copied them into the table containing my initial data so I could more easily identify matches. Included also are the classifications of the last 10 entries given the first 90 passed into the trainer.

52	Arcade Fire: Wake Up	Rock	Hip-Hop	
53	Natasha Bedingfield: Unwritten	Other	Other	
54	Coldplay: Clocks	Rock	Rock	
55	Dandy Warhols: We Used To Be Friends	Other	Other	
56	Fall Out Boy: Thnks Fr Th Mmrs	Rock	Other	
57	Flaming Lips: Fight Test	Rock	Rock	
58	Ben Folds: You Don't Know Me	Rock	Rock	
59	Franz Ferdinand: Take Me Out	Rock	Rock	
60	Frou Frou: Let Go	Other	R&B	
61	Jay-Z: Problems	Hip-Hop	Pop	
62	Jack Johnson: Flake	Other	Pop	
63	Junior Senior: Move Your Feet	Rock	Hip-Hop	
64	Kings of Leon: Sex On Fire	Rock	Pop	
65	Lady Gaga: Poker Face	Pop	Rock	
66	LCD Soundsystem: Daft Punk Is Playing At My House	Pop	Rock	
67	MIMS: This Is Why I'm Hot	Hip-Hop	Rock	
68	Nada Surf: Always Love	Pop	Rock	
69	Phoenix: Lisztomania	Other	Pop	
70	Rihanna feat Jay-Z: Umbrella	Hip-Hop	Hip-Hop	
71	Rilo Kiley: Portions For Foxes	Rock	Rock	
72	Andy Samberg's SNL Digital Shorts	Rock	Hip-Hop	
73	The Shins: Caring Is Creepy	Rock	Rock	
74	Britney Spears: Toxic	Pop	Rock	
75	Spoon: I Turn My Camera On	Other	Hip-Hop	
76	Ting Tings: Shut Up and Let Me Go	Rock	Rock	
77	Usher feat Ludacris & L'il Jon: Yeah!	Hip-Hop	Rock	
78	Vicious Vicious: Shake That Ass On the Dancefloor	Pop	Rock	
79	White Stripes: Seven Nation Army	Rock	Rock	
80	X-Press feat David Byrne: Lazy	Other	Hip-Hop	
81	"Weird Al" Yankovic: White 'N' Nerdy	Hip-Hop	Rock	
82	Daniel Powter: Bad Day	Rock	Rock	
83	The All-American Rejects: Gives You Hell	Other	Rock	
84	Aqualung: Brighter Than Sunshine	Pop	Rock	
85	India Arie: Video	R&B	Rock	
86	Matt Pond PA: Halloween	Rock	Other	
87	MIA: Paper Planes	Hip-Hop	Other	
88	My Morning Jacket: Golden	Rock	Rock	
89	Bloc Party: Banquet	Rock	Rock	
90	Loretta Lynn: Portland, Oregon	Other	Rock	
91	New Pornographers: The Bleeding Heart Show	Rock	Rock	
92	Ray LaMontagne: You Are the Best Thing	Other	Other	Other
93	Owl City: Fireflies	Pop	Rock	Rock
94	AR Rahman: Jai Ho	Pop	Rock	Hip-Hop
95	Santogold: LES Artistes	Rock	Rock	Rock
96	Scissor Sisters: Take Your Mama	Pop	Rock	Rock
97	Jill Scott: Golden	R&B	Rock	Rock
98	Shakira: Whenever, Wherever	Other	Hip-Hop	Rock
99	Soulja Boy: Crank That (Soulja Boy)	Hip-Hop	Rock	Rock
100	Sufjan Stevens: The Man of Metropolis Steals Our Hearts	Rock	Rock	Rock
101	Yellowcard: Ocean Avenue	Pop	Rock	Rock

Bryan Carey

CS432 Assignment #9

Due: 11:59pm May 1 2017

The table with the calculated values with the first 50 used as training data is as follows. Formulas were taken from Dr. Nelson's lecture slides, per usual.

Categories	True +	False +	False -	Precision	Recall	F-Measure
Pop	0	4	10	0	0	0
Rock	15	16	6	0.483871	0.714286	0.57692308
Hip-Hop	1	6	6	0.142857	0.142857	0.14285714
R&B	1	1	2	0.5	0.333333	0.4
Other	3	3	6	0.5	0.333333	0.4
Averages				0.325346	0.304762	0.30395604

3. Repeat question #2, but use the first 90 entries to train your classifier and the last 10 entries for testing.

I ran the trainer same as before, except this time I replaced line 30 with 'if count < **trainingCount90:**'

The output is stored in Classify10.txt and are also included in the table above for visual reference. The updated table is as follows.

Categories	True +	False +	False -	Precision	Recall	F-Measure
Pop	0	0	4	0	0	0
Rock	2	6	0	0.25	1	0.4
Hip-Hop	0	1	1	0	0	0
R&B	0	0	1	0	0	0
Other	1	0	1	1	0.5	0.66666667
AVERAGES				0.25	0.3	0.21333333

Bryan Carey

CS432 Assignment #9

Due: 11:59pm May 1 2017

The questions below is for 5 points extra credit

==========

4. Rerun question 3, but with "10-fold cross validation". What was the change, if any, in precision and recall (and thus F-Measure)?

For this portion of the assignment I modified my existing trainer.py file and once again used Dr. Nelson's slides with the assistance of a classmate. CrossValidation.py once again called from PCI's docclass.

```
input_1 = open('dataTitles.txt', 'r')
input_2 = open('dataClassification.txt', 'r')
input_3 = open('combinedsumtitle2.txt', 'r')
titles=[]
categories=[]
summaryTitles=[]

for entry in input_1:
   titles.append(entry)
for entry in input_2:
   categories.append(entry)
for entry in input_3:
   summaryTitles.append(entry)

count = 0
count2 = 0

#n is used as a constraint for the range of items cross validated, when n = 2
#items 11-20 will be cross validated
n = 10

filename = "CrossValidation" + str(n*10)+".txt"
outfile = open(filename,"wb")
classifier = docclass.fisherclassifier(docclass.getwords)

classifier.setdb('mln.db')

summariesToBeClassified = summaryTitles[((n-1)*10):(n*10)]
subList_1 = summaryTitles[:(n-1)*10]
subList_2 = summaryTitles[(n)*10:]
trainingdata_Entries = subList_1 + subList_2
categoriesToBeClassified = categories[((n-1)*10):(n*10)]
subList_3 = categories[:(n-1)*10]
subList_4 = categories[(n)*10:]
trainingdata_Categories= subList_3+subList_4

while count < 90:
    print count
    docclass.train(classifier, trainingdata_Entries[count], trainingdata_Categories[count])
    count += 1
count =0

while count2<10:
   print count2
   print summariesToBeClassified[(count2)]
   prediction = classifier.classify(summariesToBeClassified[(count2)])
   outfile.write(prediction)
   count2+=1
```

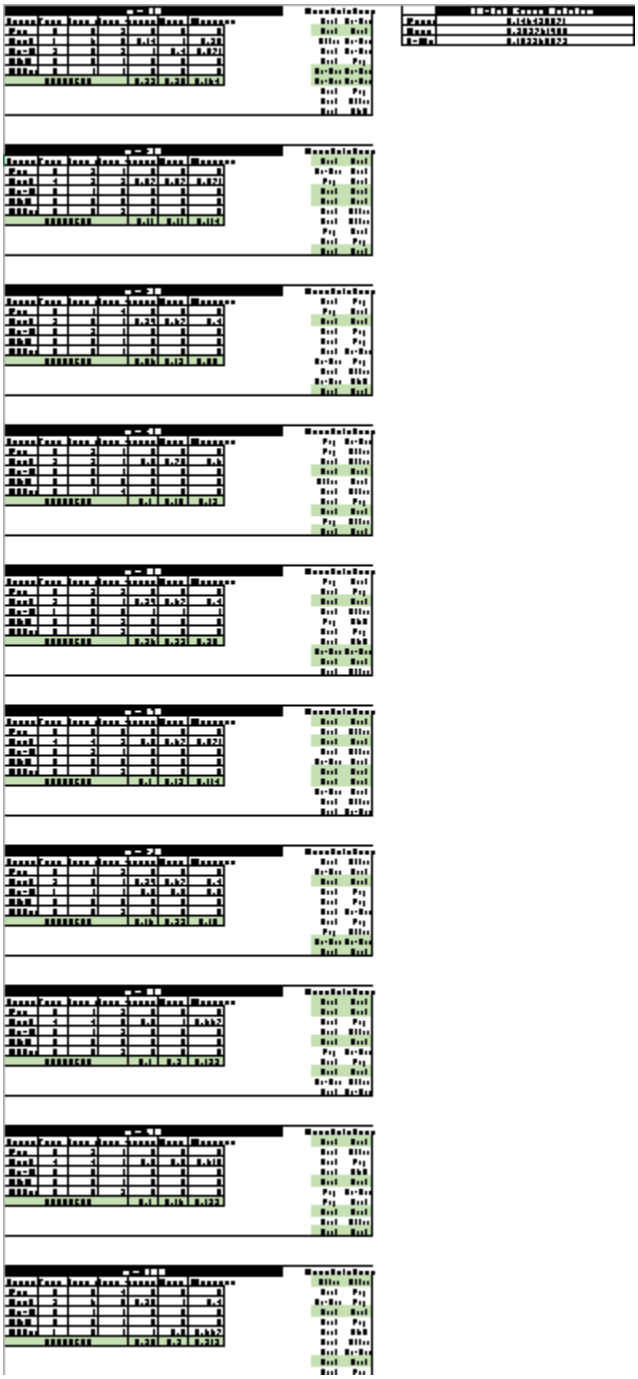
Bryan Carey

CS432 Assignment #9

Due: 11:59pm May 1 2017

I ran the program a total of ten times, each time replacing the value of n [1-10] each time. N in each occurance was used to select a specific range of items- for example when n = 2, the output of the program will contain the second set of 10 items and their classification(11-20).

I placed the results in a spreadsheet following the previous format. Once the all 10 tables were complete, averaged the 10 Macro-Averages to yield the ten-fold cross validation.



Bryan Carey

CS432 Assignment #9

Due: 11:59pm May 1 2017

Here is a close up of the first entry as well as the final answer computed.

n = 10						
Categories	True +	False +	False -	Precision	Recall	F-Measure
Pop	0	0	2	0	0	0
Rock	1	6	0	0.142857	1	0.25
Hip-Hop	2	0	3	1	0.4	0.57142857
R&B	0	0	1	0	0	0
Other	0	1	1	0	0	0
AVERAGES				0.228571	0.28	0.16428571

Results:	Actual:
Rock	Hip-Hop
Rock	Rock
Other	Hip-Hop
Rock	Hip-Hop
Rock	Pop
Hip-Hop	Hip-Hop
Hip-Hop	Hip-Hop
Rock	Pop
Rock	Other
Rock	R&B

	10-Fold Cross Validation
Precision	0.146428571
Recall	0.203761905
F-Measure	0.152260073

Below is a comparison of the average values computed

	50	10	10-Fold Validation
Accuracy	0.325346	0.25	0.146428571
Recall	0.304762	0.3	0.203761905
F-Measure	0.30395604	0.21333333	0.152260073

It appears the percentage decreased among methods, contrary to my belief. The disparity between 50 & 10 trained data supports the analogy that Dr. Nelson used in class with the basketball players altering the predictions when added to a height pool during the last lecture of the semester.

SOURCES

All graphs can be found in their respective questions, saved as PDFs.

https://github.com/arthur-e/Programming-Collective-Intelligence/blob/master/chapter6/docclass.py

https://docs.python.org/2/library/re.html

http://stackoverflow.com/questions/13748674/python-re-sub-how-to-substitute-all-u-or-us-with-you

http://www.dummies.com/programming/big-data/data-science/data-science-cross-validating-in-python/